

**SISTEM DETEKSI DINI PENCURIAN MOTOR BERBASIS
INTERNET OF THINGS (IOT) DENGAN ARDUINO DAN
MODUL GSM
SKRIPSI**

TEKNIK ELEKTRO KONSENTRASI REKAYASA KOMPUTER

Ditujukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



CHATARINA ILLINOSKA BULUAMA
NIM. 145060307111016

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2021

LEMBAR PENGESAHAN
SISTEM DETEKSI DINI PENCURIAN MOTOR BERBASIS *INTERNET OF THINGS* (IOT) DENGAN ARDUINO DAN MODUL GSM
SKRIPSI
TEKNIK ELEKTRO KONSENTRASI REKAYASA KOMPUTER

Ditujukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



CHATARINA ILLINOSKA BULUAMA

NIM. 145060307111016

Skripsi ini telah direvisi dan disetujui oleh dosen pembimbing
pada tanggal 25 Juli 2021

Dosen Pembimbing I.

Dosen Pembimbing II.

Waru Djuriatno, S.T., M.T.
NIP. 19690725 199702 1 001

Dr. Ir. Muhammad Aswin, M.T.
NIP. 19640626 199002 1 001

Mengetahui,

Ketua Jurusan Teknik Elektro



Muhammad Aziz Muslim, S.T., M.T., Ph.D.
NIP. 19741203 200012 1 001

JUDUL SKRIPSI

SISTEM DETEKSI DINI PENCURIAN MOTOR BERBASIS *INTERNET
OF THINGS* (IOT) DENGAN ARDUINO DAN MODUL GSM

Nama Mahasiswa : Chatarina Illinoska Buluama

NIM : 145060307111016

Program Studi : Teknik Elektro

Konsentrasi : Rekayasa Komputer

Dosen Pembimbing 1 : Waru Djuriatno, S.T., M.T.



Dosen Pembimbing 2 : Dr. Ir. Muhammad Aswin, M.T.



Tim Dosen Penguji :

Dosen Penguji 1 : Adharul Muttaqin, S.T., M.T.

 disetujui tanggal 25 Juli 2021

Dosen Penguji 2 : Dr. Ir. Bambang Poerwadi, M.S.

 disetujui tanggal 25 Juli 2021

Tanggal Ujian : 21 Juli 2021

SK Penguji : No. 1251 Tahun 2021

"In the midst of winter, I finally learned that there was in me an invincible summer." – Albert Camus



*Teriring ucapan terima kasih kepada :
Mama tercinta*

PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas di dalam Naskah Skripsi ini adalah asli dari pemikiran saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Bekasi, 22 Juli 2021

Mahasiswa,

Materai 6000

Tanda tangan

CHATARINA ILLINOSKA BULUAMA
NIM. 145060307111016



RINGKASAN

Chatarina Illinoska Buluama, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Juli 2021, Sistem Deteksi Dini Pencurian Motor Berbasis *Internet of Things* (IoT) dengan Arduino dan Modul GSM, Dosen Pembimbing : Waru Djuriatno dan Muhammad Aswin.

Implementasi *Internet of Things* (IoT) dapat digunakan untuk mengurangi kerugian waktu yang dialami pemilik motor ketika motornya dicuri. Penelitian ini bertujuan untuk merancang dan mengembangkan sistem yang mampu mendeteksi secara dini gejala pencurian motor dan secara instan mengirimkan notifikasi ke ponsel pengguna ketika memang terdeteksi. Gejala pencurian motor yang dimaksud meliputi tiga parameter sederhana yang masukannya diambil dari tiga sensor berbeda, yakni adanya gerakan di depan lubang kunci motor, adanya tekanan pada jok motor, serta adanya pergerakan motor dari kondisi diam. Sistem juga dirancang untuk dapat menampilkan status keamanan motor serta posisi terakhir motor pada aplikasi Android sederhana. Status keamanan motor diproses oleh Arduino dan dikirimkan menggunakan modul GSM ke layanan penyimpanan data di internet yang kemudian dikirimkan ke aplikasi Android di ponsel pengguna. Hasil pengujian fungsionalitas sistem pada penelitian ini belum sesuai dengan perancangan dikarenakan adanya kegagalan pemrosesan data yang mengakibatkan terganggunya proses pengiriman data oleh sistem.

Kata kunci : *Internet of Things; Modul GSM; Arduino; Pencurian Motor*

SUMMARY

Implementation of Internet of Things (IoT) can be utilized to decrease the loss of time experienced by a motorcycle owner when their motorcycle got stolen. This research aims to design and develop a system that can detect early symptoms of motorcycle theft and instantly sends a notification to the user's cellphone when it is detected. The symptoms of motorcycle theft that was mentioned translates to three simple parameters which inputs are taken from three different sensors, the parameters being detected movement in front of the motorcycle keyhole, detected pressure on the motorcycle seat, and whether or not the motorcycle moved from a still condition. The system is also designed to display the motorcycle's security status and its last location on a simple Android application. The motorcycle's security statuses are processed by an Arduino and are sent through GSM module to a cloud storage, which are then sent to the Android application in the user's cellphone. Results of system functionality tests in this research are yet to be in accordance with the design as there were problems in processing the data which resulted in failure of sending the data.

Keywords : *Internet of Things; GSM Module; Arduino; Motorcycle Theft*



PENGANTAR

Dengan menyebut nama Yesus yang maha pengasih lagi maha penyayang. Segala puji bagi Tuhan yang Maha Kuasa, yang telah memberikan rahmat berupa kesehatan, kesempatan, kemampuan dan nikmat lain yang tak terkira sehingga penulis mampu menyelesaikan skripsi yang berjudul “Sistem Deteksi Dini Pencurian Motor Berbasis *Internet of Things* (IoT) dengan Arduino dan Modul GSM” sebagai salah satu syarat untuk menyelesaikan Program Sarjana (S1) Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya

Penulis menyadari bahwa untuk menyelesaikan masa studi dan skripsi ini, tidak lepas dari bantuan, bimbingan, dan dukungan dari berbagai pihak. Pada kesempatan kali ini, penulis ingin menyampaikan rasa terima kasih yang sebesar – besarnya kepada:

1. Tuhan yang Maha Kuasa, yang telah memberikan rahmat berupa kesehatan, kesempatan, kemampuan, dan nikmat lain yang tak terkira.
2. Bapak Mathias Buluama dan Mama Veronica Shinta Ambarsari yang selalu memberikan doa, dukungan, semangat, kasih sayang, perjuangan, serta motivasi dalam setiap langkah dalam hidup hingga dapat memberikan kesempatan kepada anak pertama mereka. Serta Adik Angelica Vermariska Dwi Hapsari dan Adik Maria Virginia Ruth yang juga selalu memberikan semangat dan dukungan untuk menjadi lebih baik lagi hingga dapat menyelesaikan masa studi. Semoga penulis dapat membalas apa yang telah diberikan oleh keluarga.
3. Bapak Prof. Ir. Hadi Suyono, S.T., M.T., Ph.D., IPU, ASEAN. Eng. selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya yang telah memberikan banyak nasehat dan pengarahan dalam mengembangkan prestasi akademik dan organisasi di Jurusan Teknik Elektro.
4. Ibu Ir. Nurussa'adah, M.T. selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya yang telah banyak memberikan nasehat dan pengarahan dalam mengembangkan prestasi akademik dan organisasi di Jurusan Teknik Elektro.
5. Ibu Rahmadwati, S.T., M.T., Ph.D. selaku Ketua Program Studi S1 Jurusan Teknik Elektro Universitas Brawijaya.
6. Bapak Adharul Muttaqin, S.T., M.T. selaku KKJF Rekayasa Komputer dan Dosen Pembimbing Akademik yang telah memberikan waktu, ilmu, pengalaman, saran, kritik, dan motivasi mulai dari pertemuan pertama di ruang kerja beliau sampai dengan penyusunan skripsi.
7. Bapak Waru Djuriatno, S.T., M.T. selaku Dosen Pembimbing I yang telah memberikan bimbingan, ilmu, saran, dan motivasi selama penyusunan skripsi.
8. Bapak Dr. Ir. Muhammad Aswin, M.T. selaku Dosen Pembimbing II yang telah memberikan bimbingan, ilmu, saran, dan motivasi selama penyusunan skripsi.
9. Segenap dosen, tenaga kependidikan, dan staff Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.
10. Sahabat-sahabat cicak tercinta Adyanto Trinugroho, Enna “pantijompoenthusiast” Wibowo dan Anastasia Meicy yang selalu hadir, mendukung, dan menyemangati di saat-saat tergelap penulis dengan penuh cinta dan penerimaan.
11. Kak Dode atas bantuannya dan dukungannya yang luar biasa pada saat proses pengerjaan skripsi.
12. Om Iwan atas terapi dan dukungannya yang telah membantu penulis melewati masa-masa terkelam penulis sebelum akhirnya bisa memulai proses penulisan skripsi.
13. Ikhwan Fajri, Iskandar Zulkarnain, dan Elang Timur yang telah menemani dan

- mendukung penulis selama masa perkuliahan dan penulisan skripsi.
14. Farra, Winda, Ical, dan anggota Males Multichat lainnya yang selalu mendukung penulis dalam proses penyelesaian skripsi.
 15. Teman – teman dari server Discord LoR, Keia, Enver, Rafu, Wan, Ryu, Ben, dan lain-lain yang telah menemani penulis pada saat proses penulisan skripsi dan selalu memberikan dukungan moral.
 16. Teman – teman dari server Discord Nirdharma, Dragondwagon, Shira, ptonthedestroyer, Peacefire, Lockhart77, Saurki, Potat Kuru, RifqyDGamer, Junichi, Phem, Volarevia, Alpaccaz, Eksos, dan lain-lain yang tidak dapat penulis sebutkan satu per satu yang telah menemani penulis pada saat proses penulisan skripsi dan selalu memberikan dukungan moral.
 17. Teman – teman komunitas DnD, Duchess Andromeda, Joaru, dan Zee yang telah mendukung penulis pada saat proses penulisan skripsi.
 18. Teman – teman di gedung himpunan Teknik Elektro dan ruangan RisTIE yang selalu membuat penulis merasa diterima di semester-semester akhir masa perkuliahan.
 19. Teman – teman Teknik Elektro angkatan 2014 sesama pejuang skripsi yang selalu berbagi ilmu dan saling menyemangati di akhir masa studi.
 20. Semua pihak yang telah memberikan doa, bantuan, serta dukungan baik secara langsung maupun tidak langsung dalam penyusunan skripsi ini yang namanya tidak bisa saya sebutkan satu persatu.

Bekasi, 22 Juli 2021

Penulis

DAFTAR ISI

PENGANTAR.....	i
DAFTAR ISI.....	iii
DAFTAR GAMBAR.....	v
DAFTAR TABEL.....	vii
DAFTAR LAMPIRAN.....	viii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Penelitian.....	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Sistematika Penyelesaian Masalah.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 <i>Internet of Things (IoT)</i>	5
2.2 Arduino.....	5
2.3 Modul GSM/GPRS.....	6
2.3.1 AT Command.....	7
2.3.2 SIM800L.....	8
2.4 Passive Infrared Sensor (PIR Sensor).....	10
2.2.1 HC-SR501.....	12
2.5 Force Sensitive Resistor (FSR Sensor).....	13
2.6 MPU-6050.....	15
2.4.1 Pin-Out Board GY-521.....	15
2.7 NEO-6MV2.....	16
2.7 Buzzer.....	17
2.8 Arduino IDE.....	18
2.9 ThingSpeak.....	18
2.10 MIT App Inventor.....	20
BAB III METODE.....	22
3.1 Gambaran Umum Sistem.....	22
3.2 Spesifikasi Sistem.....	23
3.3 Perancangan Mekanika Deteksi Gejala Pencurian Motor.....	24
3.3.1 Sensor PIR, Sensor FSR, dan Akselerometer.....	24

3.3.2	Modul GSM	25
3.3.3	Modul GPS	25
3.3.4	Alarm	25
3.3	Perancangan Sistem Komunikasi Arduino dengan Android Pengguna	25
3.3.1	Komunikasi Modul GSM dengan ThingSpeak	26
3.3.2	Komunikasi ThingSpeak dengan Android Pengguna	26
3.4	Perancangan Aplikasi Android	27
3.4.1	Rancangan Antarmuka Aplikasi Android	27
3.4.2	Rancangan Proses Kerja Aplikasi Android	31
3.5	Proses Kerja Sistem	34
3.6	Pengujian Sistem	38
3.7	Pengambilan Kesimpulan dan Saran	39
BAB IV	HASIL DAN PEMBAHASAN	57
4.1	Pengujian Performa Sensor PIR	57
4.1.1	Prosedur Pengujian	57
4.1.2	Hasil Pengujian dan Analisis	58
4.2.1	Prosedur Pengujian	59
4.2.2	Hasil Pengujian dan Analisis	59
4.3	Pengujian Performa Akselerometer	60
4.3.1	Prosedur Pengujian	60
4.3.2	Hasil Pengujian dan Analisis	61
4.4	Pengujian Akurasi Modul GPS	61
4.4.1	Prosedur Pengujian	62
4.4.2	Hasil Pengujian dan Analisis	62
4.5	Pengujian Sistem Komunikasi Data Secara Keseluruhan	63
4.5.1	Prosedur Pengujian	63
4.5.2	Hasil Pengujian dan Analisis	64
4.6	Pengujian Sistem Secara Keseluruhan	65
4.6.1	Prosedur Pengujian	65
4.6.2	Hasil Pengujian dan Analisis	66
BAB V	KESIMPULAN DAN SARAN	67
5.1	Kesimpulan	67
5.2	Saran	67
	DAFTAR PUSTAKA	68
	LAMPIRAN	70

DAFTAR GAMBAR

Gambar 2.1 Arduino Uno Rev3 (Sumber: store.arduino.cc).....	6
Gambar 2.2 SIM800L tampak depan (Sumber: lastminuteengineers.com)	8
Gambar 2.3 SIM800L tampak belakang (Sumber: lastminuteengineers.com).....	8
Gambar 2.4 Pin out SIM800L (Sumber: lastminuteengineers.com).....	9
Gambar 2.5 Area deteksi PIR sensor (Sumber: learn.adafruit.com).....	11
Gambar 2.6 Sinyal keluaran PIR sensor (Sumber: learn.adafruit.com).....	11
Gambar 2.7 Sensor IR (Sumber: learn.adafruit.com).....	12
Gambar 2.8 Lensa Fresnel (Sumber: BHlens.com)	12
Gambar 2.9 Foto makro lensa Fresnel (Sumber: learn.adafruit.com).....	13
Gambar 2.10 Tampak depan dan belakang FSR (Sumber: learn.adafruit.com)	14
Gambar 2.11 Lapisan FSR (Sumber: learn.adafruit.com).....	14
Gambar 2.12 FSR persegi panjang (Sumber: phidgets.com)	15
Gambar 2.13 Board GY-521(Sumber: electrosome.com).....	15
Gambar 2.14 Modul NEO-6MV2 (Sumber: components101.com).....	16
Gambar 2.15 Buzzer (Sumber: components101.com)	17
Gambar 2. 16 Antarmuka Arduino IDE	18
Gambar 2. 17 Contoh tampilan data berupa grafik pada ThingSpeak	19
Gambar 2. 18 API Key yang disediakan oleh ThingSpeak	20
Gambar 2. 19 Antarmuka MIT App Inventor bagian “Design”	21
Gambar 2. 20 Antarmuka MIT App Inventor bagian “Blocks”	21
Gambar 3.1 Diagram blok umum dari sistem yang dirancang	22
Gambar 3.2 Skematik sistem.....	24
Gambar 3. 3 Alur sistem komunikasi Arduino, ThingSpeak, dan aplikasi pada Android pengguna	26
Gambar 3. 4 Halaman utama rancangan antarmuka aplikasi android.....	27
Gambar 3. 5 Halaman “Lokasi Motor” pada rancangan antarmuka aplikasi android.....	29
Gambar 3. 6 Contoh rancangan antarmuka aplikasi android ketika status keseluruhan “aman”.....	30
Gambar 3. 7 Contoh rancangan antarmuka aplikasi android ketika status keseluruhan “waspada”	30
Gambar 3. 8 Contoh rancangan antarmuka aplikasi android ketika status keseluruhan “bahaya”	31
Gambar 3. 9 Blok diagram proses kerja aplikasi	31
Gambar 3. 10 Diagram alir pemrosesan data yang masuk ke aplikasi.....	32
Gambar 3.11 Blok diagram proses kerja sistem	35
Gambar 3.12 Blok pengambilan data.....	36
Gambar 3.13 Blok pemrosesan dan pengiriman data.....	36
Gambar 3.14 Blok alarm dan notifikasi	37
Gambar 3.15 Blok tindakan.....	37
Gambar 3.16 Blok request data	38
Gambar 4. 1 Tampilan serial monitor pada saat fungsi pengiriman data dijalankan	64

Gambar 4. 2 Tampilan serial monitor ketika keseluruhan fungsi pengiriman dan permintaan data dijalankan pada sistem.....



DAFTAR TABEL

Tabel 2. 1 Beberapa perintah AT command dan fungsinya	7
Tabel 3. 1 Kombinasi status individual yang membentuk status keseluruhan	33
Tabel 4. 1 Hasil pengujian performa sensor PIR	58
Tabel 4. 2 Hasil pengujian performa sensor FSR	59
Tabel 4. 3 Hasil pengujian performa akselerometer	61
Tabel 4. 4 Hasil pengujian keakurasian modul GPS	62
Tabel 4. 5 Kondisi pada setiap sensor dan status pada aplikasi	66



DAFTAR LAMPIRAN

Lampiran 1. Foto Alat	70
Lampiran 2. Skematik Breadboard Sistem	71
Lampiran 3. <i>Sourcecode</i> Arduino	71
Lampiran 4. Blok pemrograman pada MIT App Inventor: Screen 1	85
Lampiran 5. Blok Pemrograman pada MIT App Inventor: Screen 2	87



BAB I PENDAHULUAN

1.1 Latar Belakang

Salah satu sarana bepergian yang paling populer di Indonesia adalah sepeda motor. Berdasarkan data Badan Pusat Statistik (BPS) Kota Malang per Juli 2017, jumlah sepeda motor tercatat ada 441.123 unit pada 2015 dan naik menjadi 456.693 unit pada 2016 (BPS Kota Malang, 2017). Namun, dari data yang dirilis Polres Malang Kota dan Polres Malang selama Januari–September 2017, tercatat ada 625 kasus motor hilang (Jawa Pos Radar Malang, 2017), dan sejumlah 33.526 kasus selama tahun 2017 di seluruh Indonesia (BPS Subdirektorat Statistik Politik dan Keamanan, 2019). Tidak hanya itu, dari 625 kasus yang tercatat di kota Malang, hanya 139 kasus yang terungkap (Jawa Pos Radar Malang, 2017).

Kurangnya sistem peringatan dan pengawasan untuk pencegahan pencurian sepeda motor tidak hanya menimbulkan kerugian materiil, namun juga merugikan waktu pemilik kendaraan yang bersangkutan sampai pihak kepolisian menemukan kendaraan tersebut. Oleh karena itu dibutuhkan sebuah alat yang dapat mendeteksi secara dini gejala pencurian motor dan memperingatkan pemilik sepeda motor agar ia dapat segera menghampiri motornya atau segera menghubungi pihak kepolisian agar dapat menemukan motor tersebut lebih cepat.

Di era digital ini, penggunaan internet telah menjadi suatu kebutuhan. Menurut data Kemkominfo dan Bappenas, pada tahun 2010 penggunaan internet oleh masyarakat mencapai angka 45 juta orang pengguna dan 2,7 juta orang pelanggan, dan pada tahun 2012 jumlah tersebut telah mencapai 65 juta orang pengguna. Masyarakat mengakses internet dengan beragam perangkat dan teknologi, termasuk ponsel. (Ruth, 2013) Maka, alangkah lebih praktis apabila pendeteksian pencurian motor dari alat yang dimaksud dapat dikirimkan ke ponsel pengguna alat tersebut melalui koneksi internet menggunakan sistem *Internet of Things* (IoT).

Dalam penelitian ini akan dirancang sebuah sistem yang akan memperingatkan pemilik motor melalui ponselnya ketika motornya mengalami gejala pencurian. Gejala pencurian ditandai dengan adanya aktivitas di depan lubang kunci motor, adanya tekanan pada jok motor, dan atau adanya pergerakan motor.

Dengan menggunakan Arduino yang dihubungkan dengan modul GSM, sistem akan mengirimkan peringatan kepada pemilik motor dalam bentuk SMS. Pada saat itu juga alarm yang ada pada sistem akan menyala sehingga akan memperingatkan orang-orang di sekitar.

Pengguna dapat mengawasi kondisi keamanan motor melalui aplikasi yang terhubung ke internet, kemudian pengguna dapat mematikan alarm melalui aplikasi. Selanjutnya modul GPS pada sistem akan mengirimkan lokasi motor kepada pemilik motor. Dengan adanya sistem ini diharapkan dapat meminimalisir kerugian waktu yang dialami oleh pemilik motor ketika motornya telah dicuri.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan diatas maka dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana rancangan sistem deteksi dini gejala pencurian motor berbasis IoT?
2. Bagaimana cara mendeteksi secara dini gejala pencurian motor menggunakan Arduino?
3. Bagaimana cara mengkomunikasikan kondisi keamanan motor ke pengguna menggunakan Modul GSM dan aplikasi Android?
4. Bagaimana cara melacak posisi motor pengguna melalui aplikasi Android?
5. Bagaimana cara memberi peringatan kepada pengguna dan orang di sekitar ketika motor mengalami gejala pencurian motor?

1.3 Batasan Penelitian

Adapun batasan-batasan masalah pada penelitian ini adalah sebagai berikut:

1. Kendaraan bermotor yang digunakan dalam penelitian ini adalah sepeda motor merk Honda Beat Pop.
2. Indikasi gejala pencurian motor dibatasi pada adanya gerakan manusia di depan kunci sepeda motor, adanya tekanan pada jok motor, dan adanya pergerakan motor.
3. Terbatas untuk pengguna ponsel dengan sistem operasi Android.
4. Pengujian dilakukan dengan melihat keberhasilan alat mendeteksi gejala pencurian motor dan pengiriman notifikasi berupa SMS serta kondisi keamanan dan posisi motor ke ponsel pengguna.
5. Alat selalu berada dalam posisi *standby* namun baru akan mendeteksi gejala pencurian motor dan mengirimkan status keamanan motor setelah sistem dinyalakan melalui aplikasi.
6. Alat tidak bekerja jika modul GSM dan modul GPS tidak mendapatkan sinyal.
7. Arduino diprogram dengan Arduino IDE.

8. Keseluruhan alat diletakkan di dalam bagasi motor, kecuali sensor PIR yang disematkan di bawah jok di area kaki pengemudi motor dan dikamufleskan menggunakan bahan berwarna hitam.

1.4 Tujuan

Tujuan dari penelitian ini adalah:

1. Mengembangkan sistem pendeteksi dini gejala pencurian motor yang mampu mempersingkat waktu diketahuinya pencurian motor dari saat motor masih mengalami gejala pencurian motor.
2. Mengembangkan sistem pengawasan keamanan motor yang dapat dipantau dengan mudah melalui satu aplikasi Android oleh pengguna.
3. Mengembangkan sistem pencegahan pencurian kendaraan bermotor yang menggunakan Arduino maupun modul GSM.

1.5 Manfaat

Penelitian ini diharapkan dapat membantu meminimalisir kerugian waktu yang dialami pengguna motor ketika motornya dicuri serta dapat dijadikan referensi untuk rancangan sistem keamanan secara publik termasuk namun tidak terbatas pada sepeda motor.

1.6 Sistematika Penyelesaian Masalah

Dalam proses penyelesaian masalah akan dilakukan beberapa metode, antara lain:

1. Studi literatur

Studi literatur dilakukan untuk mencari informasi yang berkaitan dengan penelitian ini dan untuk membandingkan dengan hasil penelitian yang sudah pernah dilakukan sebelumnya. Studi literatur dilakukan dengan memahami teori-teori pendukung penelitian. Teori-teori ini didapatkan dari berbagai sumber referensi buku, internet, maupun jurnal penelitian lainnya.

2. Simulasi dan Perancangan Sistem

Simulasi dilakukan untuk mendapatkan hasil yang terbaik yang nantinya akan diimplementasikan kedalam perancangan sistem.

3. Pengujian

Pengujian dilakukan untuk memastikan bahwa sistem berjalan dengan baik dan untuk mencari kekurangan dari sistem agar dapat disempurnakan lagi.

4. Penyusunan laporan dan Pengambilan Kesimpulan
Bertujuan untuk melaporkan hasil dari setiap pelaksanaan dari penelitian dan mengambil kesimpulan dari setiap analisis yang telah dilakukan



BAB II TINJAUAN PUSTAKA

Bab ini memaparkan literatur, jurnal, dan teori dasar yang digunakan sebagai acuan dan pedoman dalam proses penyelesaian masalah yang ada pada pelaksanaan penelitian ini.

2.1 *Internet of Things (IoT)*

Internet of things (IoT) adalah sebuah konsep dalam dunia komputasi di mana objek-objek fisik terhubung ke jaringan internet dan mampu memperkenalkan dirinya ke perangkat lainnya. Internet of things merupakan suatu paradigma baru yang sedang populer dalam skenario komunikasi nirkabel modern. (Atzori, Iera, & Morabito, 2010) Ide dasar dari konsep ini adalah keberadaan berbagai macam *thing* atau objek yang tersebar di sekitar kita, seperti *Radio Frequency IDentification* (RFID), sensor, aktuator, ponsel, dan lain-lain, yang melalui skema pengalamatan yang unik, dapat berinteraksi satu sama lain dan bekerja sama untuk mencapai tujuan yang sama.

Kekuatan utama IoT adalah dampaknya yang sangat besar pada aspek-aspek kehidupan sehari-hari dan perilaku calon pengguna. Dari sudut pandang pengguna pribadi, efek paling jelas dari pengenalan IoT dapat terlihat dalam lingkup pekerjaan maupun kehidupan rumah tangga.

Potensi yang ditawarkan oleh IoT membuat pengembangan pengaplikasian yang berjumlah besar menjadi mungkin, di mana dari pengaplikasian tersebut hanya sedikit yang saat ini tersedia dalam kehidupan kita. Lingkungan-lingkungan dalam kehidupan sehari-hari seperti di rumah, saat bepergian, saat sakit, saat bekerja, saat berolahraga, saat ini hanya dilengkapi dengan objek-objek dengan kecerdasan primitif, kebanyakan tanpa kemampuan komunikasi. Dengan IoT, kita dapat memberikan objek-objek ini kemungkinan untuk berkomunikasi satu sama lain dan untuk menjelaskan informasi yang dapat diambil dari lingkungan sekitar.

2.2 *Arduino*

Arduino merupakan *platform open source* baik secara perangkat keras maupun perangkat lunak. Arduino terdiri dari mikrokontroler ATmega328 dengan menggunakan kristal osilator 16 MHz. Catu daya yang disarankan untuk mencatu sistem Arduino berkisar antara 7-12V DC. Port Arduino seri ATmega terdiri dari 20 pin yang meliputi 14 pin I/O digital dengan 6 pin dapat berfungsi sebagai output PWM (*Pulse Width Modulation*) dan 6

pin sebagai I/O analog. Kelebihan Arduino adalah tidak membutuhkan *flash programmer* eksternal karena di dalam chip mikrokontroler Arduino telah diisi dengan *bootloader* yang membuat proses pengunggahan menjadi lebih sederhana. Untuk koneksi terhadap komputer dapat menggunakan RS232 to TTL *converter* atau menggunakan chip USB ke *serial converter* (Sulaiman, 2012). Arduino yang digunakan pada skripsi ini adalah Arduino seri Uno. Arduino Uno dilengkapi dengan *static random-access memory* (SRAM) berukuran 2KB untuk menyimpan data, *flash memory* berukuran 32KB, dan *erasable programmable read-only memory* (EEPROM) untuk menyimpan program (Kadir, 2012).



Gambar 2.1 Arduino Uno Rev3 (Sumber: store.arduino.cc)

2.3 Modul GSM/GPRS

Sebuah modul GSM atau modul GPRS merupakan sebuah chip atau rangkaian yang digunakan untuk menjalin komunikasi antara perangkat seluler atau mesin komputasi dengan sistem GSM atau GPRS. Modemnya (modulator-demodulator) adalah bagian yang sangat penting di sini. Modul-modul ini terdiri dari modul GSM atau GPRS yang diberi tenaga dari sebuah rangkaian sumber daya dan antarmuka komunikasi (seperti RS-232, USB 2.0, dan lainnya) untuk komputer. Sebuah modem GSM dapat berupa sebuah perangkat modem terdedikasi dengan koneksi serial, USB, atau Bluetooth, atau dapat juga berupa sebuah telepon seluler yang menyediakan kapabilitas GSM Modem. Modul GSM atau modul GPRS mirip dengan modem, namun ada satu perbedaan: Modem GSM/GPRS merupakan peralatan eksternal, sedangkan modul GSM/GPRS adalah sebuah modul yang bisa diintegrasikan di dalam sebuah peralatan. Modul GSM/GPRS merupakan sebuah perangkat keras tersemat. (Electronics For U, 2019)

Modul GSM/GPRS yang digunakan pada skripsi ini adalah Modul GSM SIM800L.

2.3.1 AT Command

Modul GSM/GPRS menggunakan AT command. AT command adalah instruksi yang digunakan untuk mengontrol sebuah modem, dan meliputi semua fungsionalitas telepon seluler melalui komputer seperti menelepon dan menerima telepon, SMS, MMS, dan lain-lain. AT command kebanyakan digunakan untuk servis SMS dan MMS berbasis komputer. AT command disebut demikian karena setiap baris command dimulai dengan "AT" atau "at", yang merupakan singkatan dari "Attention" atau "perhatian".

Modem GSM/GPRS mendukung set AT command yang spesifik terhadap teknologi GSM, yang termasuk command-command yang berhubungan dengan SMS seperti AT+CMGS (mengirimkan pesan SMS), AT+CMSS (mengirimkan pesan SMS dari penyimpanan), AT+CMGL (membuat daftar pesan SMS) dan AT+CMGR (membaca pesan SMS).

Perlu diperhatikan bahwa "AT" merupakan prefiks yang memberitahu modem mengenai permulaan dari sebuah baris command. "AT" itu sendiri bukanlah bagian dari nama command AT. Sebagai contoh, nama command AT pada ATD sebenarnya adalah D dan pada AT+CMGS, nama command AT sebenarnya adalah +CMGS. (Electronics For U, 2019)

Berikut adalah secuplik hal yang bisa dilakukan menggunakan AT command dengan modem GSM/GPRS atau telepon seluler:

Tabel 2.1 Beberapa perintah AT command dan fungsinya

Fungsi	Contoh	Perintah
SMS	Mengirim	AT+CMGS
	Menerima	AT+CMGR
	Menulis	AT+CMGW
	Menghapus	AT+CMGD
	Mendapatkan notifikasi dari SMS yang baru saja diterima	AT+CNMI
Mengontrol presentasi kode hasil / pesan error AT command.	Menyalakan pesan error tertentu	AT+CMEE
	Pesan error ditampilkan dalam format numerik	AT+CMEE=1
	Pesan error ditampilkan dalam format kata-kata	AT+CMEE=2

2.3.2 SIM800L

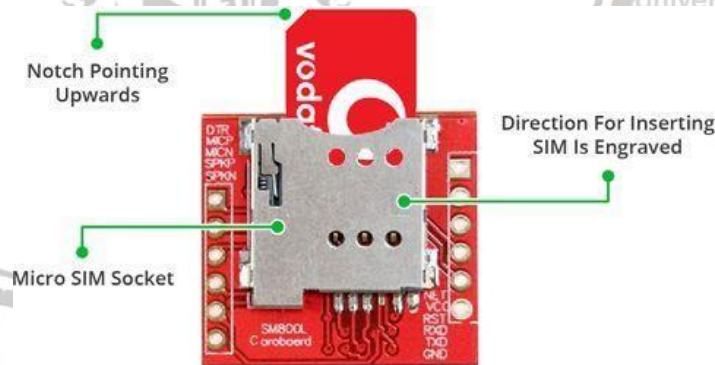
SIM800L merupakan sebuah miniatur modem GSM, yang dapat diintegrasikan ke dalam banyak proyek IoT. Jantung modul terletak pada chip seluler GSM SIM800L dari SimCom. Untuk dapat beroperasi, SIM800L membutuhkan tegangan 3.4V sampai 4.4V.



Gambar 2.2 SIM800L tampak depan (Sumber: lastminuteengineers.com)

Semua pin data yang diperlukan oleh chip GSM SIM800L dibagi menjadi pitch header 0.1", yang mencakup pin-pin yang diperlukan untuk komunikasi dengan Arduino lewat UART. Modul ini mendukung baud rate dari 1200bps hingga 115200bps dengan deteksi baud otomatis.

Modul ini membutuhkan antenna eksternal untuk dapat terhubung ke sebuah jaringan. Modul ini biasanya satu paket dengan antenna Helical yang disolder langsung pada pin NET. Boardnya juga memiliki fasilitas konektor U.FL jika ingin menggunakan antenna yang jauh dari board.



Gambar 2.3 SIM800L tampak belakang (Sumber: lastminuteengineers.com)

Di belakangnya terdapat sebuah soket SIM, yang dapat diisi dengan micro SIM card 2G apapun yang sudah teraktivasi. Berikut fitur-fitur yang dapat dilaksanakan oleh modul ini:

- Mendukung Quad-band: GSM850, EGSM900, DCS1800 dan PCS1900
- Dapat menjalin koneksi ke jaringan global GSM manapun dengan SIM 2G apapun.

- Membuat dan menerima telepon menggunakan speaker external 8Ohm dan mikrofon electret.
- Mengirim dan menerima pesan SMS.
- Mengirim dan menerima data GPRS (TCP/IP, HTTP, dll.)
- Mencari dan menerima broadcast dari radio FM.
- Kekuatan Transmisi:
 - Kelas 4 (2W) untuk GSM850
 - Kelas 1 (1W) untuk DCS1800
- Set command AT berbasis serial
- Konektor FL untuk antenna seluler
- Menerima kartu SIM Micro

Pada sisi kanan atas modul seluler SIM800L terdapat sebuah indikator LED yang akan menandakan status dari jaringan seluler. Indikator LED akan berkedip pada rentang waktu yang berbeda-beda untuk menunjukkan statusnya:

- Berkedip setiap 1 detik: Modul menyala tetapi belum terkoneksi ke jaringan seluler.
- Berkedip setiap 2 detik: Koneksi data GPRS yang diminta telah aktif.
- Berkedip setiap 3 detik: Modul telah terkoneksi ke jaringan seluler dan dapat mengirim/menerima panggilan suara dan SMS.

Karena modul SIM800L tidak menyediakan regulator tegangan, dibutuhkan sumber daya eksternal yang diatur ke tegangan 3,4V sampai 4 sampai 4,4V (idealnya 4,1V). Sumber tegangan tersebut juga harus bisa menyuplai arus 2A secara terus menerus, atau modulnya akan terus menerus mati. Dalam skripsi ini akan digunakan baterai li-po 3.7v 4000mAh.

Modul SIM800L memiliki 12 pin antarmuka. Koneksinya sebagai berikut:



Gambar 2.4 Pin out SIM800L (Sumber: lastminuteengineers.com)

1. NET adalah pin tempat menyolder antenna helical yang disediakan bersama modul.

2. **VCC** menyuplai daya untuk modul. Tegangan bisa dari 3.4V sampai 4.4V.
3. **RST (Reset)** adalah pin hard reset untuk mereset modul.
4. **RxD (Receiver)** digunakan untuk komunikasi serial.
5. **TxD (Transmitter)** digunakan untuk komunikasi serial.
6. **GND** adalah pin Ground yang harus dihubungkan dengan pin GND pada Arduino.
7. **SPK±** adalah antarmuka speaker diferensial. Dua pin speaker bisa disambungkan secara langsung pada kedua pin ini.
8. **MIC±** adalah antarmuka input mikrofon diferensial. Dua pin mikrofon bisa disambungkan secara langsung pada kedua pin ini.
9. **DTR** adalah pin yang mengaktifkan atau mematikan mode sleep. Pada kondisi high modul akan berada pada mode sleep, mematikan komunikasi serial, dan pada kondisi low modul akan kembali hidup.
10. **RING** adalah pin yang bertindak sebagai indikator Ring. Secara sederhana ia adalah pin 'interrupt' pada modul. Secara default pin ini berada pada kondisi high dan akan mengirim pulsa low selama 120ms ketika menerima sebuah panggilan. Pin ini juga bisa diatur untuk mengirimkan pulsa ketika menerima sebuah SMS. (Last Minute Engineers, n.d.)

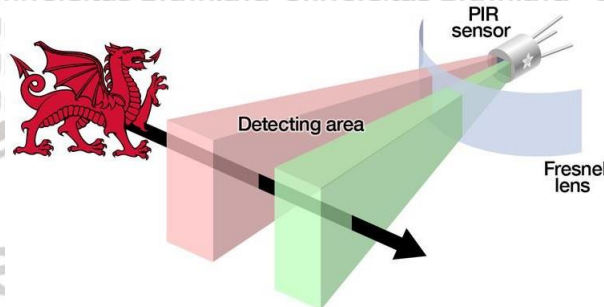
Dalam skripsi ini SIM800L dipilih karena dimensi yang kecil dan ringan sehingga mudah dipasang pada alat, serta harganya yang terbilang murah jika dibandingkan dengan modul GSM/GPRS tipe lainnya.

2.4 Passive Infrared Sensor (PIR Sensor)

Passive Infrared Sensor (PIR sensor) adalah sebuah sensor elektronik yang mengukur cahaya infrared (IR) yang terpancar dari objek-objek yang berada pada jarak pandangnya. PIR sensor mendeteksi gerakan umum, namun tidak memberikan informasi mengenai siapa atau apa yang bergerak. Kata pasif digunakan karena PIR sensor tidak memancarkan energi untuk mendeteksi, namun bekerja hanya dengan mendeteksi radiasi infrared (radiasi panas) yang dipancarkan atau terpantul dari objek.

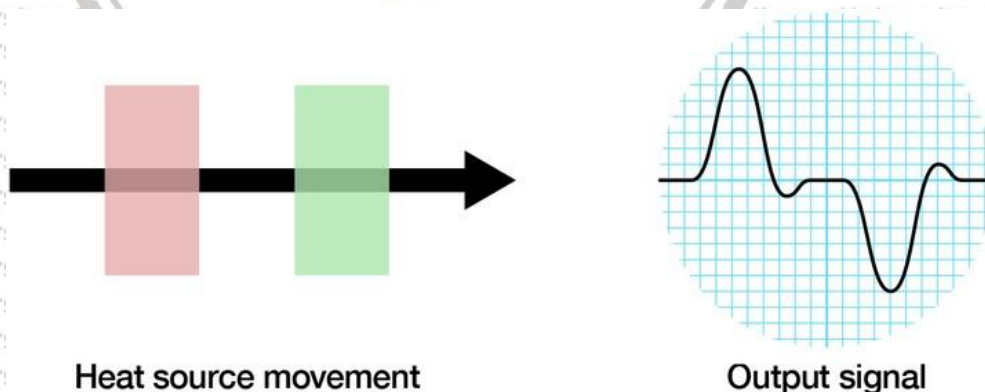
PIR sensor terdiri dari dua bagian utama: sensor piroelektrik dan sebuah lensa khusus yang disebut lensa Fresnel yang memfokuskan sinyal infrared pada sensor piroelektrik. Sensor piroelektrik memiliki dua slot persegi panjang di dalamnya yang terbuat dari bahan

yang membuat radiasi infrared dapat lewat. Di belakangnya, ada dua elektroda sensor infrared yang berbeda, satu berfungsi untuk membuat keluaran positif dan satunya lagi untuk membuat keluaran negatif. Alasannya karena untuk mendeteksi gerakan, yang dicari adalah perubahan tingkatan IR dan bukan tingkatan IR yang sama. Kedua elektroda tersebut dipasang supaya mereka saling membatalkan sinyal satu sama lain. Jika salah satu elektroda melihat lebih banyak atau lebih sedikit radiasi IR daripada elektroda lainnya, keluarannya akan berubah menjadi *high* atau *low*. Pada gambar ini bisa kita lihat bahwa kedua slot bisa ‘melihat’ melewati sekian jarak. (Adafruit, 2014)



Gambar 2.5 Area deteksi PIR sensor (Sumber: learn.adafruit.com)

Ketika sensornya diam, kedua slot mendeteksi jumlah IR yang sama, jumlah dari sekelilingnya yang teradiasi dari ruangan atau dinding ataupun luar ruangan. Ketika ada sebuah tubuh yang hangat seperti manusia atau hewan melewatinya, tubuh itu akan pertama-tama tertangkap salah satu dari sensor PIRnya, yang akan mengakibatkan perbedaan diferensial positif di antara keduanya. Ketika tubuh yang hangat tersebut meninggalkan area penglihatan, kebalikannya terjadi, di mana sensor membuat perubahan diferensial negatif. Pulsa perubahan ini adalah apa yang dideteksi. (Adafruit, 2014)

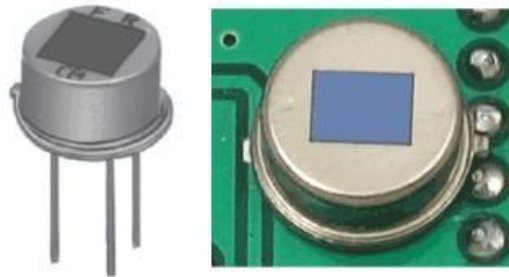


Gambar 2.6 Sinyal keluaran PIR sensor (Sumber: learn.adafruit.com)

Pada skripsi ini, PIR sensor yang akan digunakan adalah PIR Sensor model HC-SR501.

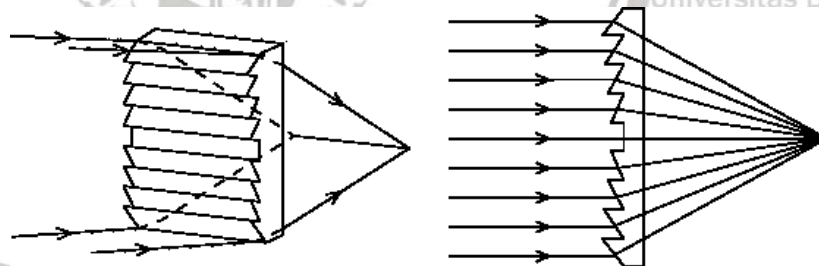
2.2.1 HC-SR501

Sensor IR pada PIR sensor model HC-SR501 diletakkan di dalam kaleng logam kedap udara untuk meningkatkan imunitas suara, temperatur, dan kelembaban. Ada sebuah jendela yang terbuat dari bahan transmisif infrared, biasanya silikon yang dilapisi, yang melindungi elemen sensornya (Adafruit, 2014).



Gambar 2.7 Sensor IR (Sumber: learn.adafruit.com)

Pada HC-SR501, lensanya hanyalah sepotong plastik, namun jika hanya sepotong plastik biasa, itu artinya area deteksinya hanya dua persegi panjang. Untuk mendapatkan area yang lebih besar, lensa yang digunakan adalah lensa sederhana yang biasa ditemukan di dalam sebuah kamera: lensa-lensa tersebut memadatkan area yang besar (seperti sebuah pemandangan) menjadi area yang kecil (dalam film atau sensor CCD). Lensa Fresnel memadatkan cahaya, menyediakan jarak pandang IR yang lebih besar kepada sensor.



Gambar 2.8 Lensa Fresnel (Sumber: BHLens.com)

Namun, karena sebetulnya ada dua sensor, dan lebih penting lagi, kita tidak menginginkan dua area deteksi berbentuk persegi panjang besar, melainkan beberapa area kecil yang tersebar, maka lensanya dibagi menjadi beberapa bagian, di mana setiap bagian adalah lensa Fresnel. (Adafruit, 2014)



Gambar 2.9 Foto makro lensa Fresnel (Sumber: learn.adafruit.com)

Faset-faset dan sub-lens yang berbeda menciptakan berbagai area deteksi, bersipisan satu sama lain. Itulah mengapa bagian tengah lens pada faset-faset di atas ‘tidak konsisten’ – setiap yang lainnya menunjuk ke salah satu dari elemen sensor PIR. (Adafruit, 2014)

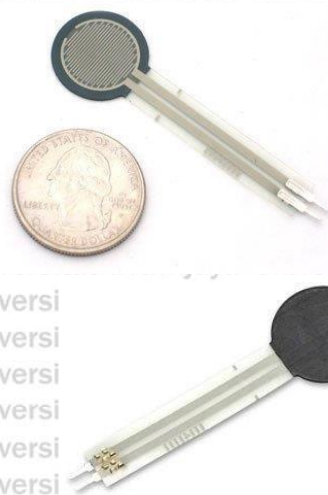
Spesifikasi HC-SR501 adalah sebagai berikut:

- Tegangan: 4.5-20V DC
- Output sinyal: digital (3.3V DC HIGH; 0V DC LOW)
- Jarak deteksi: 3-7m
- Waktu delay: 5-200s
- Sudut area deteksi: 140°
- Diameter lensa: 23mm
- Ukuran board: 32x24mm

Sensor ini dipilih karena memiliki sudut area deteksi yang cukup besar, dimensi yang kecil dan ringan sehingga mudah dipasang pada alat, serta harganya yang terbilang murah.

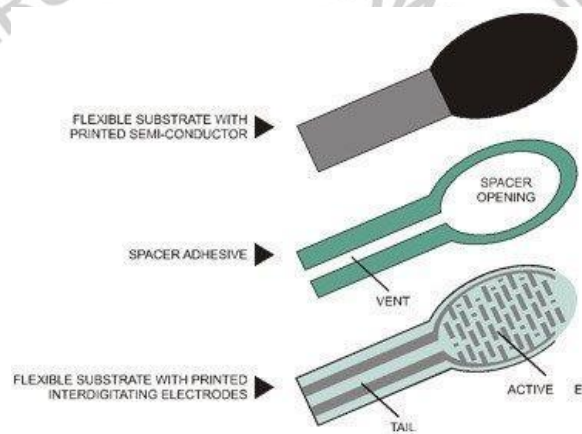
2.5 Force Sensitive Resistor (FSR Sensor)

FSR adalah sebuah bahan yang resistansinya akan berubah ketika suatu gaya, tekanan, atau stres mekanik diterapkan pada bahan tersebut. Ia adalah sebuah sensor yang bisa mendeteksi tekanan fisik, tindihan, dan berat. Teknologi ini pertama kali ditemukan dan dipatenkan pada tahun 1977 oleh Franklin Eventoff (Adafruit, 2012). Di bawah ini ada foto FSR model Interlink 402. Bagian yang sensitif adalah bagian bulat berdiameter setengah inci.



Gambar 2.10 Tampak depan dan belakang FSR (Sumber: learn.adafruit.com)

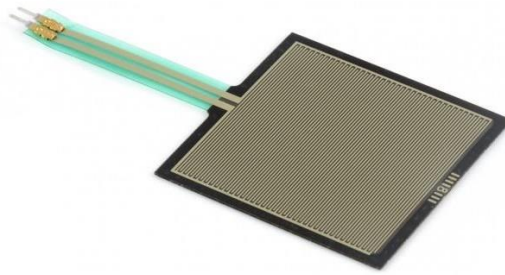
FSR terbuat dari 2 buah lapisan yang dipisahkan oleh sebuah spacer. Semakin besar tekanannya, semakin banyak titik elemen aktif yang menyentuh semikonduktor, dan hal itu membuat resistansinya menurun.



Gambar 2.11 Lapisan FSR (Sumber: learn.adafruit.com)

FSR secara sederhana adalah sebuah resistor yang mengubah nilai tahanannya (dalam ohm) tergantung seberapa keras ia ditekan. Sensor ini murah dan mudah digunakan, namun tidak akurat. Jadi ketika menggunakan FSR kita hanya bisa berharap untuk mendapatkan respon dalam bentuk kisaran. Meskipun FSR bisa mendeteksi berat, FSR tidak cocok digunakan untuk mendeteksi berapa kilogram tepatnya berat yang dideteksi. Namun, untuk pengaplikasian yang sensitif terhadap sentuhan seperti “apakah ini telah dipencet atau ditekan dan kira-kira seberapa keras”, cukup cocok untuk menggunakan FSR. (Adafruit, 2012)

Pada skripsi ini digunakan FSR dengan bentuk persegi panjang yang lebih besar dengan harapan dapat memperluas area deteksi tekanan pada jok motor.



Gambar 2.12 FSR persegi panjang (Sumber: phidgets.com)

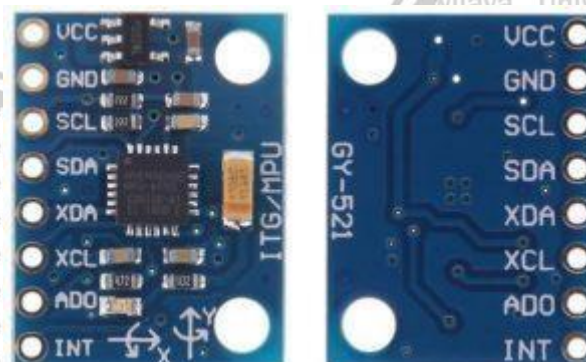
Spesifikasi Square FSR ini adalah sebagai berikut:

- Panjang: 88mm
- Lebar: 43.7mm
- Tebal: 0.42mm
- Berat: 1.12g

2.6 MPU-6050

MPU-6050 adalah sebuah chip yang dibuat oleh InvenSense yang menggabungkan akselerometer 3 sumbu dan giroskop 3 sumbu dengan *motion processor* digital yang terintegrasi di dalam board. Di dalamnya terdapat juga sensor temperatur tersemat dan oscillator di dalam chip. Chip MPU-6050 ini sangat akurat dan terdiri dari perangkat keras ADC (*analog to digital converter*) untuk setiap kanal, sehingga bisa menangkap kanal x, y, dan z pada saat yang bersamaan. Arduinonya sendiri bisa diantarmukakan dengan bus I2C. (Santos, 2018)

2.4.1 Pin-Out Board GY-521



Gambar 2.13 Board GY-521 (Sumber: electrosome.com)

- VCC : board GY-521 memiliki regulator tegangan tersemat, maka dapat diberi tegangan 5v atau 3.3v.

- GND : pin Ground
- SCL : Serial Clock Line untuk I2C
- SDA : Serial Data Line untuk I2C
- XDA : Auxiliary data
- XCL : Auxiliary clock
- AD0 : ketika pin ini diset menjadi low, alamat I2C pada board akan menjadi 0x68, ketika diset menjadi high, alamat I2C menjadi 0x69.
- INT : Interrupt keluaran digital

2.7 NEO-6MV2

Modul NEO-6MV2 adalah sebuah modul GPS (Global Positioning System) dan digunakan untuk navigasi. Modul ini memeriksa lokasinya di atas bumi dan menyediakan data keluaran berupa bujur dan lintang dari posisinya. (Components 101, 2018)



Gambar 2.14 Modul NEO-6MV2 (Sumber: components101.com)

Modul ini memiliki 4 pin keluaran. Penyalan modul dan antarmuka komunikasi dilakukan melalui 4 pin ini:

- VCC : pin daya positif
- RX: pin penerima UART
- TX: pin pengirim UARTTX
- GND : Ground

Fitur-fitur yang terdapat pada modul ini adalah:

- Penerima GPS yang dapat berdiri sendiri
- Teknologi anti-macet
- Antarmuka UART pada pin keluaran (Bisa menggunakan SPI, I2C, dan USB dengan menyolder pin ke core chip)
- Tipe penerima: 50 kanal – GPS frekuensi L1 - (WAAS, EGNOS, MSAS, GAGAN)
- Waktu fix pertama kali: Cold Start 32s, Warm Start 23s, Hot Start <1s

- Rasio update navigasi maksimum : 5Hz
- Baud rate default : 9600bps
- EEPROM dengan backup baterai
- Sensitivitas : -160dBm
- Tegangan Suplai : 3.6V
- Arus DC maksimum pada keluaran apapun : 10mA
- Limit operasi : Gravitasi-4g, Ketinggian-50000m, Kecepatan-500m/s
- Kisaran tempetatur beroperasi: -40°C sampai 85°C

Pada tugas akhir ini modul NEO-6MV2 dipilih karena efisiensinya, akurasi, serta harganya yang relatif murah. Selain itu, dimensinya yang kecil memudahkan untuk dipasang pada alat.

2.7 Buzzer

Buzzer adalah komponen yang kecil namun efisien untuk menambahkan fitur suara ke dalam sebuah proyek/sistem. Ukurannya sangat kecil dan hanya memiliki 2 pin. Ada dua tipe buzzer yang umumnya tersedia. Yang pertama adalah buzzer sederhana yang akan mengeluarkan satu nada secara terus menerus, sedangkan yang kedua adalah buzzer yang lebih besar dan bisa mengeluarkan bunyi yang terputus-putus karena adanya rangkaian osilasi internal di dalamnya. Yang akan digunakan di dalam skripsi ini adalah tipe yang pertama.

Buzzer ini dapat digunakan hanya dengan menyalakannya menggunakan sumber daya DC dari 4V hingga 9V. Bisa juga menggunakan baterai 9V, namun disarankan menggunakan sumber daya DC terregulasi +5V atau +6V. Umumnya buzzer ini diasosiasikan dengan rangkaian switch untuk menyalakan atau mematikan buzzer pada waktu dan interval tertentu. (Components 101, 2017)



Gambar 2.15 Buzzer (Sumber: components101.com)

Buzzer dapat dinyalakan dengan kedua pin berikut ini:

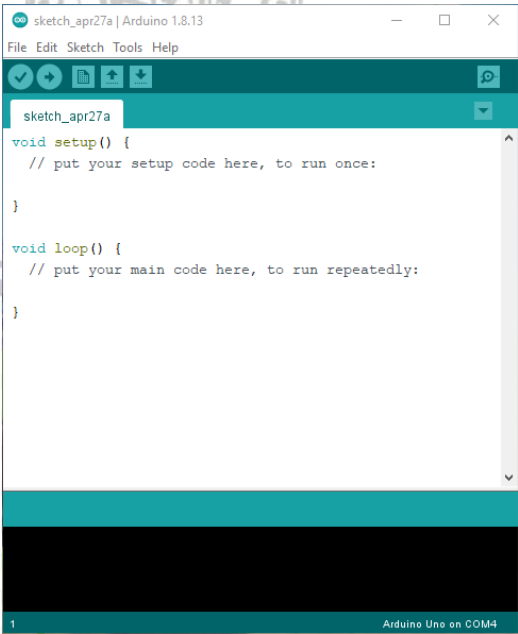
- Positif: ditandai dengan tanda + pada bagian atas buzzer. Disambungkan dengan sumber daya 5V.
- Negatif: ditandai dengan terminal yang lebih pendek. Umumnya dihubungkan ke ground pada rangkaian.

Fitur-fitur yang terdapat pada buzzer ini adalah:

- Tegangan tertera: 6V DC
- Tegangan operasi: 4-8V DC
- Arus tertera: <30mA
- Tipe suara: Bunyi bip panjang
- Frekuensi resonansi: ~2300 Hz

2.8 Arduino IDE

Arduino IDE (*Integrated Development Environment*) merupakan sebuah perangkat lunak *cross-platform* (untuk Windows, macOS, Linux) yang digunakan untuk menulis dan mengunggah program ke board Arduino yang kompatibel, atau, dengan bantuan *third-party core*, board pengembangan dari vendor lainnya. Source code untuk IDEnya sendiri dirilis di bawah lisensi GNU General Public License versi ke dua. Arduino IDE mendukung Bahasa pemrograman C dan C++ menggunakan peraturan-peraturan struktur kode tertentu. (Wikipedia, 2021)



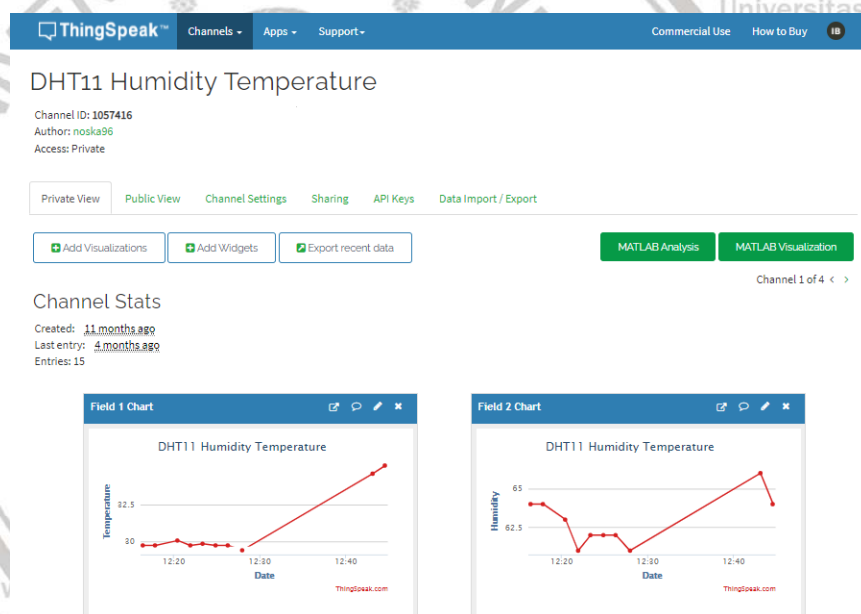
Gambar 2.16 Antarmuka Arduino IDE

2.9 ThingSpeak

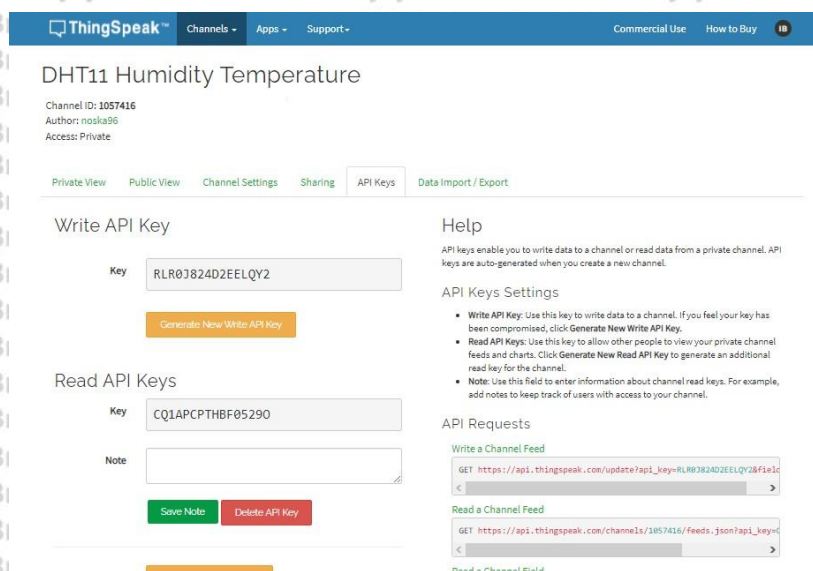
ThingSpeak merupakan sebuah aplikasi IoT dan API *open source* untuk menyimpan dan mengambil data menggunakan protokol HTTP dan MQTT melalui Internet atau melalui LAN. ThingSpeak membantu pengembangan aplikasi untuk mencatat data sensor, aplikasi pelacak lokasi, dan *social network of things* dengan pembaruan status. (Wikipedia, 2020)

ThingSpeak menyediakan visualisasi instan dari data yang dikirimkan oleh perangkat-perangkat pengguna ke ThingSpeak. Data-data ini kemudian dapat dianalisis secara online karena ThingSpeak dapat mengeksekusi kode MATLAB. (ThingSpeak, 2021)

Untuk memulai menggunakan ThingSpeak sebagai cloud tempat penyimpanan dan pengambilan data, pengguna harus membuat akun, kemudian membuat kanal baru. Satu kanal mampu menyimpan hingga 8 *field* data yang ditampilkan secara visual berupa grafik. Untuk membaca dan menuliskan data dari dan ke kanal tersebut disediakan API Key untuk digunakan oleh *thing* serta tautan untuk menuliskan dan membaca data secara manual dari browser.



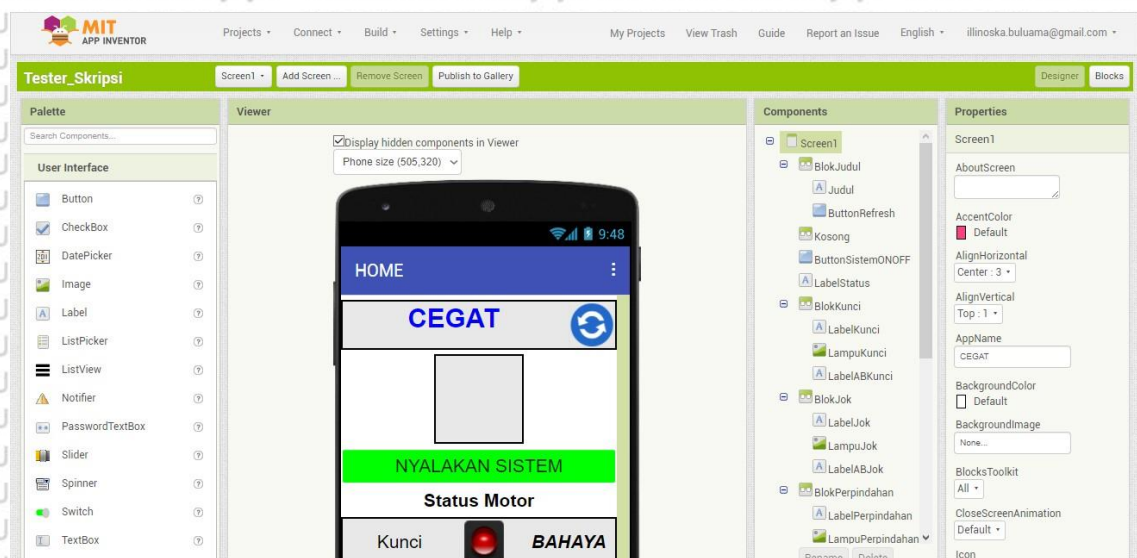
Gambar 2. 17 Contoh tampilan data berupa grafik pada ThingSpeak



Gambar 2. 18 API Key yang disediakan oleh ThingSpeak

2.10 MIT App Inventor

MIT App Inventor adalah aplikasi web IDE *open source* yang pada awalnya disediakan oleh Google dan sekarang dikelola oleh Massachusetts Institute of Technology (MIT). Aplikasi yang dibuat menggunakan MIT App Inventor dapat digunakan pada dua sistem operasi, yaitu Android dan iOS. (Wikipedia, 2021) Pengembangan aplikasi dilakukan melalui antarmuka grafis yang intuitif, di mana metode yang digunakan adalah *drag and drop* objek visual pada bagian “*Design*” untuk membentuk antarmuka aplikasinya, kemudian *drag and drop* blok-blok yang tersedia pada bagian “*Blocks*” untuk membentuk programnya. Aplikasi yang dikembangkan dapat langsung disimulasikan secara *real time* menggunakan aplikasi MIT AI2 Companion yang dapat diunduh di Google Play Store dan App Store. Apabila blok-blok yang disediakan oleh aplikasi web MIT App Inventor dianggap kurang memadai, pengembang dapat menambahkan ekstensi untuk memperbanyak fungsionalitas yang dapat digunakan.



Gambar 2. 19 Antarmuka MIT App Inventor bagian “Design”

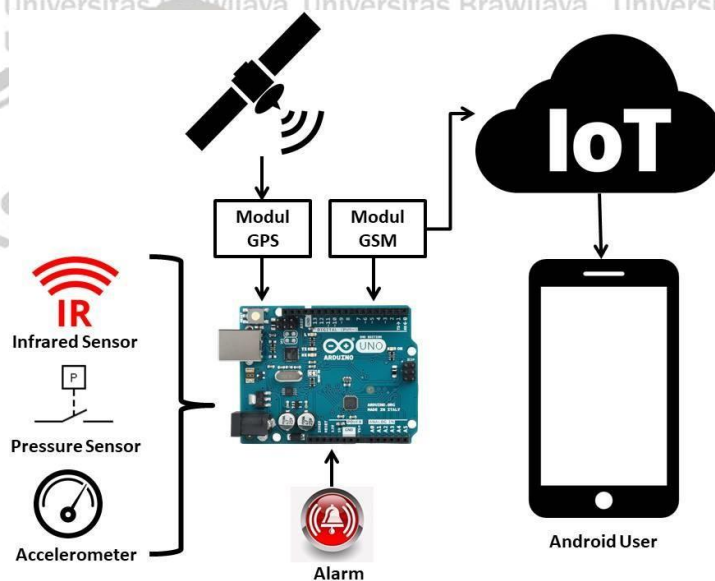


Gambar 2. 20 Antarmuka MIT App Inventor bagian “Blocks”

BAB III METODE

Pada bab ini akan dibahas mengenai gambaran umum sistem, spesifikasi sistem, perancangan mekanika pendeteksian sistem, perancangan komunikasi sistem, perancangan aplikasi Android, proses kerja sistem, cara pengujiannya, serta pengambilan kesimpulan dan saran.

3.1 Gambaran Umum Sistem



Gambar 3.1 Diagram blok umum dari sistem yang dirancang

Skema deteksi dini gejala pencurian sepeda motor dimulai dengan pengambilan data menggunakan sensor infrared, sensor tekanan, dan akselerometer. Data-data tersebut selanjutnya diolah oleh Arduino untuk menentukan apakah motor sedang mengalami gejala pencurian motor. Apabila sistem mendeteksi positif terjadi gejala pencurian motor, yakni, salah satu dari sensor mengirimkan sinyal positif, maka Arduino akan mengirimkan notifikasi berupa SMS ke Android pengguna menggunakan modul GSM. Alarm pada sistem yang diletakkan di bagasi motor akan menyala, kemudian pengguna dapat melihat status keamanan motor yang dikirimkan ke aplikasi Android pengguna melalui IoT. Setelah itu pengguna dapat memutuskan apabila ia ingin mematikan alarm dan mengetahui posisi motor. Posisi motor dapat diketahui dengan mengaktifkan modul GPS pada Arduino melalui aplikasi. Skema keseluruhan digambarkan dalam Gambar 3.1.

Sistem yang dirancang diharapkan mampu:

1. Mendeteksi adanya gerakan di depan lubang kunci motor, mendeteksi adanya tekanan pada jok motor, dan mendeteksi adanya pergerakan motor.
2. Menyalakan alarm ketika sistem mendeteksi positif terjadi gejala pencurian motor.
3. Menghubungi pengguna melalui kiriman notifikasi berupa SMS.
4. Menunjukkan status motor ketika pengguna membuka aplikasi, yang mencakup: status keamanan kunci motor, status jok motor, status pergerakan motor, serta status secara keseluruhan.
5. Mengirimkan hingga 10 titik terakhir koordinat lokasi motor kepada pengguna.

3.2 Spesifikasi Sistem

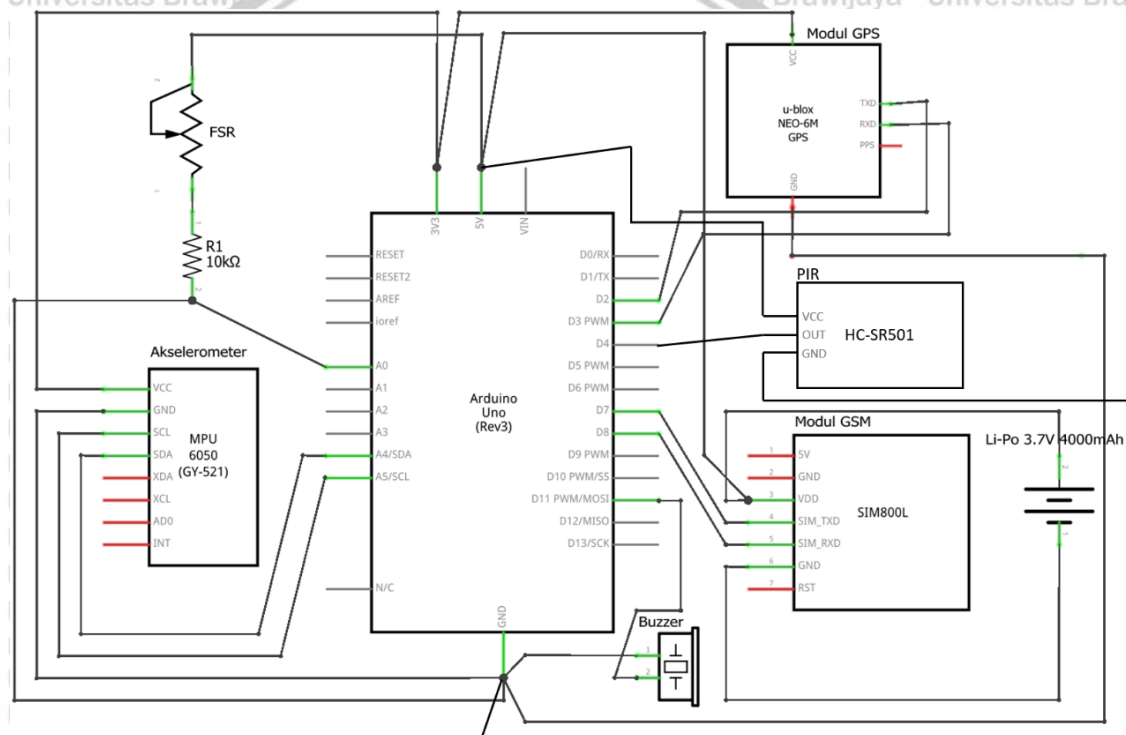
Spesifikasi sistem ditentukan dengan tujuan agar dapat membuat sistem sesuai dengan yang diharapkan. Sistem yang dirancang akan memiliki spesifikasi sebagai berikut.

1. Sensor Passive Infrared (PIR) HC-SR501 digunakan dengan tujuan mampu mendeteksi gerakan manusia di depan lubang kunci motor.
2. Sensor tekanan Force Sensitive Resistor (FSR) digunakan dengan tujuan mampu mendeteksi tekanan pada jok motor..
3. Akselerometer GY-521 MPU-6050 digunakan dengan tujuan mampu mendeteksi adanya pergerakan motor.
4. Board Arduino Uno mampu memproses data yang diambil dari kedua sensor dan akselerometer dan melakukan proses klarifikasi.
5. Modul GSM SIM800L digunakan dengan tujuan untuk menyambungkan Arduino dengan cloud ThingSpeak.
6. Modul GPS NEO-6M V2 digunakan dengan tujuan untuk menitiklokasikan posisi motor secara *real-time*.
6. Buzzer digunakan untuk output alarm sederhana. Untuk pengembangannya bisa diganti dengan alarm lain yang lebih cocok dan mumpuni.

7. ThingSpeak digunakan sebagai sarana IoT atau perantara Arduino dengan aplikasi sederhana yang akan dibuat untuk Android user.
8. Ponsel canggih dengan sistem operasi Android digunakan untuk menampilkan data beserta lokasi motor dengan jelas. Ponsel canggih yang digunakan adalah Samsung Galaxy A20 dengan Android 10.0, dengan antarmuka aplikasi sederhana.
9. MIT App Inventor digunakan sebagai alat untuk membuat aplikasi sederhana yang telah disebutkan.

3.3 Perancangan Mekanika Deteksi Gejala Pencurian Motor

Berikut merupakan skematik sistem:



Gambar 3.2 Skematik sistem

3.3.1 Sensor PIR, Sensor FSR, dan Akselerometer

PIR Sensor akan diletakkan di bawah jok motor di area kaki pengemudi, sedangkan FSR Sensor akan diletakkan di dalam busa jok motor di tempat pengemudi duduk. Akselerometer akan diletakkan bersama dengan Arduino dan seluruh modul lainnya di dalam sebuah kotak yang disembunyikan di dalam bagasi motor. Untuk pemasangan sensor PIR dan FSR yang perlu menjangkau lokasi yang

cukup jauh dari Arduino, akan disambungkan dengan kabel jumper male to female menuju breadboard yang terhubung dengan VCC dan GND pada Arduino. Arduino sendiri akan diberi sumber daya dari powerbank DC5V/1A 10000mAh. Sebagai sumber daya untuk akselerometer, digunakan output tegangan 3.3V dari Arduino yang diparalel dengan modul GPS.

3.3.2 Modul GSM

Karena Modul GSM SIM800L bekerja pada tegangan 3,7 - 4,2 Volt, maka diperlukan sumber daya eksternal dengan spesifikasi tegangan yang memenuhi syarat tersebut. Sumber daya eksternal yang dipilih adalah baterai li-po 3.6V 4000mAh. SIM800L dipasang di breadboard, kemudian pada bagian VCC dan GND dihubungkan ke baterai eksternal dengan dua buah kabel jumper. Dengan dua kabel jumper lainnya, VCC dan GND SIM800L dihubungkan ke bagian positif dan negatif breadboard yang terhubung ke pin 5V dan GND Arduino Uno. Hal ini dikarenakan komunikasi serial SIM800L mengharuskan terhubungnya pin VCC dan GND SIM800L ke Arduino.

3.3.3 Modul GPS

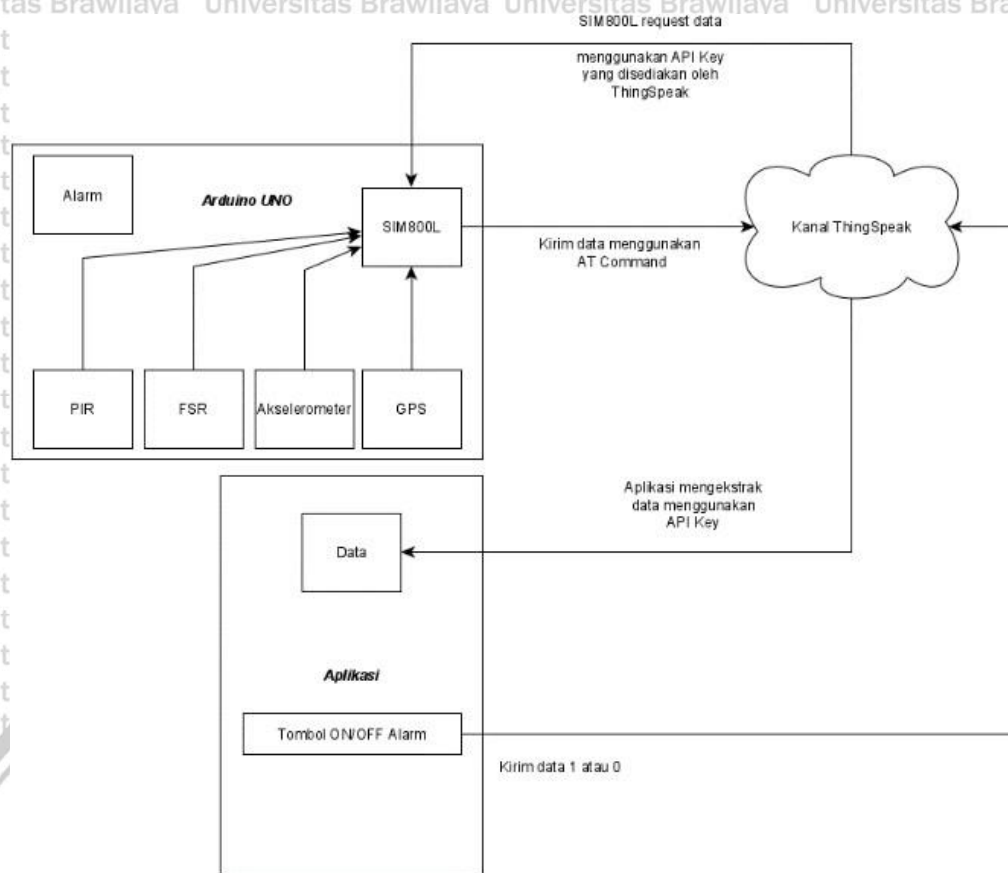
Pin VCC modul GPS akan disambungkan dengan pin 3.3V pada Arduino dan pin GND akan dihubungkan dengan kabel jumper male to female menuju breadboard yang terhubung dengan VCC dan GND pada Arduino.

3.3.4 Alarm

Alarm yang pada sistem ini berbentuk buzzer dipasang pada breadboard yang dihubungkan ke Arduino dengan kabel jumper.

3.3 Perancangan Sistem Komunikasi Arduino dengan Android Pengguna

Alur sistem komunikasi Arduino, ThingSpeak, dan aplikasi Android pengguna dapat dilihat pada Gambar 3.3.



Gambar 3. 3 Alur sistem komunikasi Arduino, ThingSpeak, dan aplikasi pada Android pengguna.

3.3.1 Komunikasi Modul GSM dengan ThingSpeak

Untuk dapat mengkomunikasikan sistem dengan ThingSpeak, pada listing program Arduino, GSM akan dihubungkan dengan API (Application Program Interface) Key dari server ThingSpeak. API Key yang digunakan adalah Write API Key. Satu API Key dapat digunakan untuk satu kanal ThingSpeak, yang dapat memuat hingga delapan field untuk delapan data yang berbeda-beda. Dari 8 field yang dapat digunakan tersebut, akan digunakan 5 field yang masing-masing akan memuat data untuk sensor PIR, sensor FSR, akselerometer, GPS, serta status alarm. Setelah sistem dinyalakan, Arduino akan terus-menerus mengirimkan data dari ketiga sensor, modul GPS, dan alarm ke ThingSpeak sehingga data di ThingSpeak akan terupdate secara kontinyu.

3.3.2 Komunikasi ThingSpeak dengan Android Pengguna

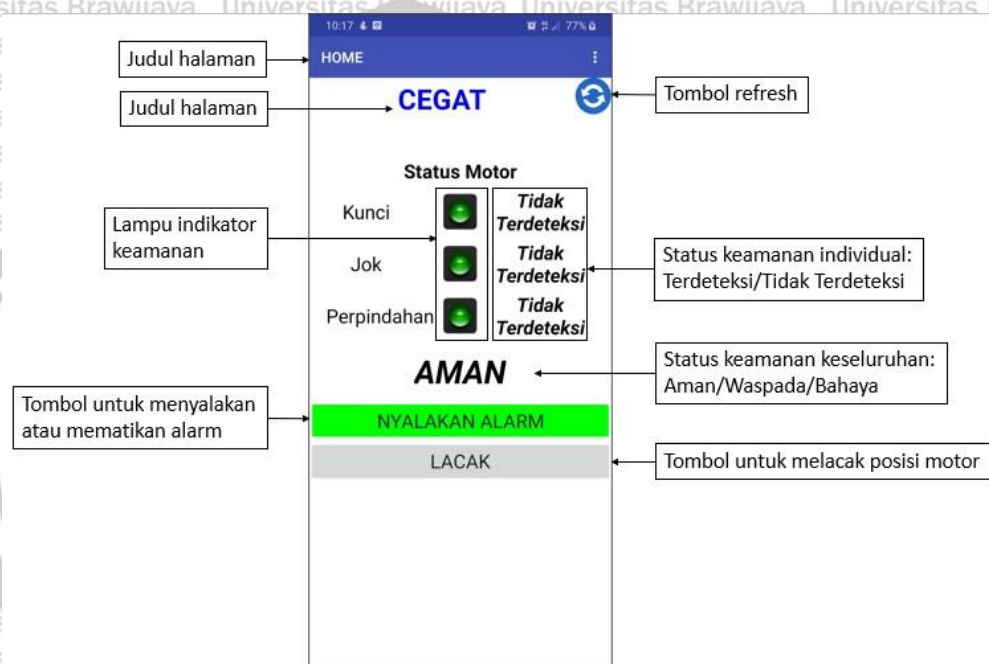
Untuk dapat mengkomunikasikan Android dengan ThingSpeak, pada blok program aplikasi akan ditambahkan API Key ThingSpeak untuk membaca data dari ThingSpeak. API Key yang digunakan adalah Read API Key. Data akan diminta

ketika aplikasi dibuka baik secara manual maupun melalui notifikasi, ketika pengguna mengetuk tombol “refresh” pada aplikasi, dan ketika pengguna mengetuk tombol “lacak” pada aplikasi. Setelahnya, data yang tampil pada aplikasi android pengguna akan sesuai dengan data pada ThingSpeak ketika data tersebut diminta.

3.4 Perancangan Aplikasi Android

3.4.1 Rancangan Antarmuka Aplikasi Android

Aplikasi Android akan diberi nama “CEGAT” sebagai kependekan dari “Cepat Tanggap Terpercaya.” Rancangan antarmuka aplikasi Android dapat dilihat pada gambar 3.4 dan 3.5.



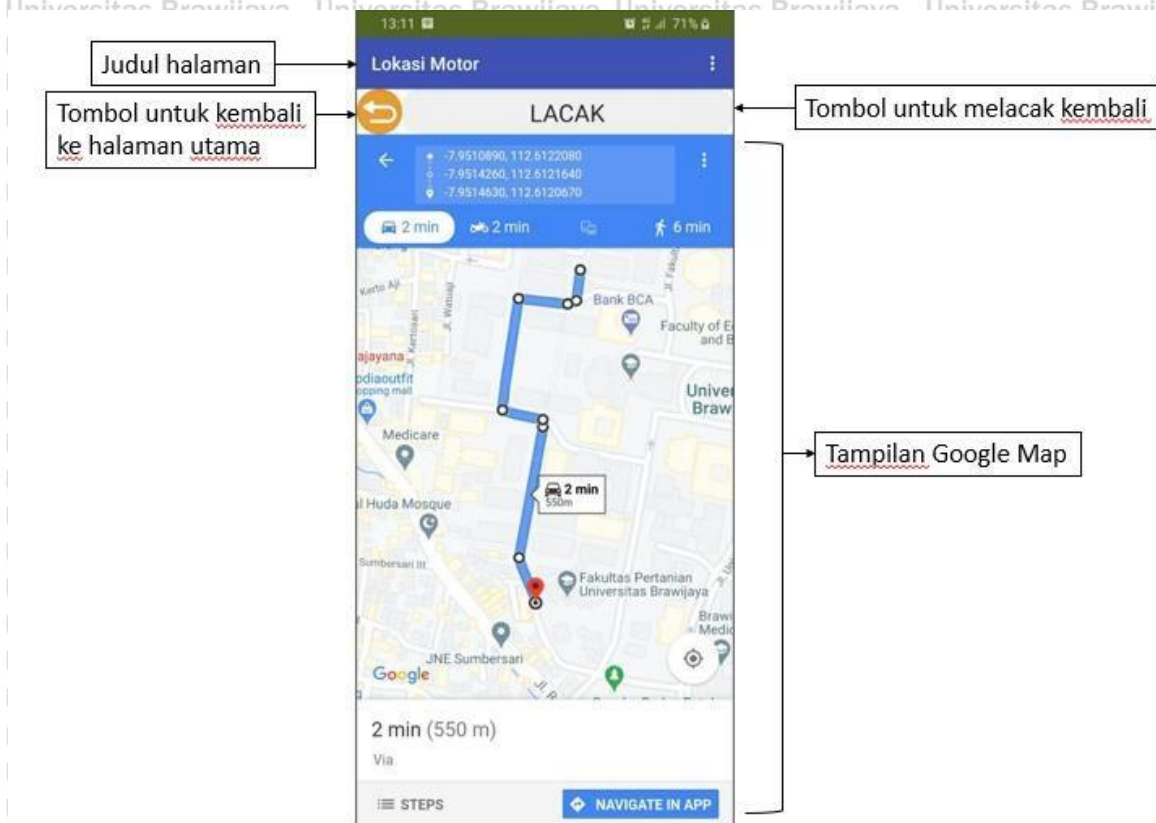
Gambar 3. 4 Halaman utama rancangan antarmuka aplikasi android

Berikut penjelasan untuk setiap bagian dari antarmuka aplikasi:

- Judul halaman: menunjukkan judul halaman. Dalam aplikasi ini hanya akan ada 2 halaman, yaitu halaman “HOME” (halaman utama), dan halaman “Lokasi Motor”.
- Nama aplikasi: menampilkan nama aplikasi. Dalam hal ini, nama aplikasi adalah “CEGAT” sebagai kependekan dari “Cepat Tanggap Terpercaya.”
- Tombol refresh: jika diketuk akan memuat ulang halaman, status individual dan status keseluruhan keamanan motor.

- d. Lampu indikator keamanan: lampu dapat menyala hijau atau merah. Hijau jika status tidak terdeteksi, dan merah jika status terdeteksi.
- e. Status keamanan individual: menunjukkan status keamanan masing-masing parameter gejala pencurian motor, yaitu status kunci, status jok, maupun status pergerakan.
 - Status “Kunci” merujuk pada ada atau tidaknya aktivitas di depan lubang kunci motor sehingga ada dua status, yaitu “tidak terdeteksi” ketika tidak ada aktivitas, atau “terdeteksi” ketika ada aktivitas di depan lubang kunci motor.
 - Status “Jok” merujuk pada ada atau tidaknya tekanan pada jok, sehingga ada dua status, yaitu “tidak terdeteksi” ketika tidak ada tekanan pada jok, atau bahaya ketika ada tekanan pada jok.
 - Status “Perpindahan” merujuk pada ada atau tidaknya pergerakan motor dari posisi diam sehingga ada dua status, yaitu “tidak terdeteksi” ketika tidak ada pergerakan motor, atau “terdeteksi” ketika ada pergerakan motor.
- f. Status keamanan keseluruhan: menunjukkan status keamanan keseluruhan motor, yang didapat dari gabungan status keamanan individual. Kombinasi status keamanan individual motor yang membentuk status keamanan keseluruhan motor akan dibahas pada sub-subbab rancangan alur kerja aplikasi. Pada status keamanan keseluruhan ini akan ada tiga status, yaitu aman, waspada, atau bahaya.
- g. Tombol untuk menyalakan atau mematikan alarm: tombol ini digunakan untuk mematikan alarm ketika alarm menyala otomatis pada status keseluruhan motor waspada atau bahaya, atau menyalakan alarm secara manual.
- h. Tombol “LACAK” untuk melacak posisi motor: jika diketuk, tombol ini akan membawa aplikasi ke halaman “Lokasi Motor” dan dengan segera menunjukkan beberapa titik lokasi terakhir motor yang akan membentuk suatu jalur dengan memanfaatkan website Google Maps. Jika pada halaman “Lokasi Motor” diketuk lagi, akan memuat ulang titik-titik lokasi terakhir motor.

- i. Tombol untuk kembali ke halaman utama: jika diketuk, akan membawa aplikasi kembali ke halaman utama (“HOME”).
- j. Tampilan Google Map: menunjukkan titik-titik lokasi terakhir motor, yang dapat dinavigasikan secara langsung menggunakan aplikasi Google Map.



Gambar 3. 5 Halaman “Lokasi Motor” pada rancangan antarmuka aplikasi android.

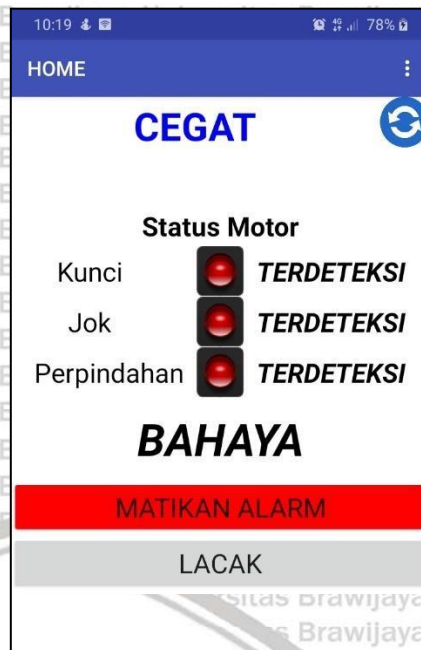
Pada gambar 3.6, 3.7, dan 3.8 dapat dilihat contoh rancangan antarmuka aplikasi android pada saat sistem sudah diaktifkan.



Gambar 3. 6 Contoh rancangan antarmuka aplikasi android ketika status keseluruhan “aman”.



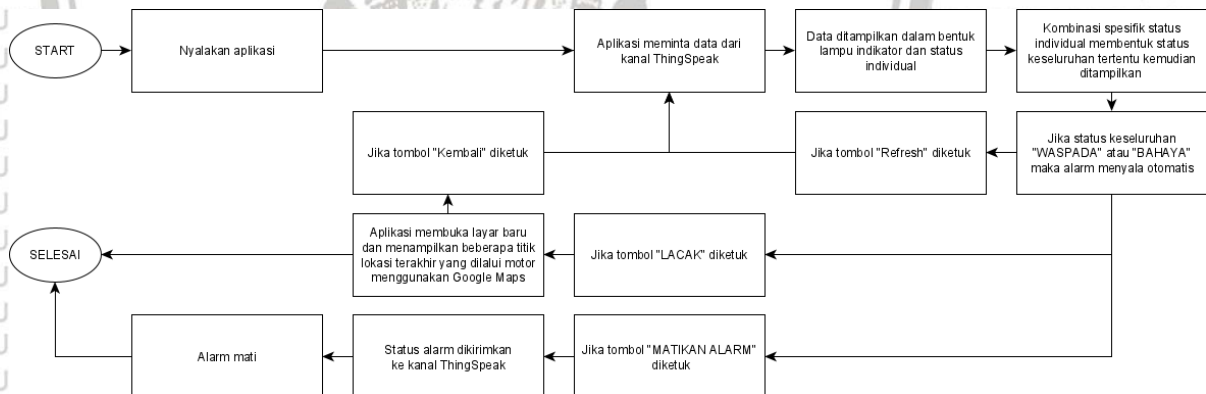
Gambar 3. 7 Contoh rancangan antarmuka aplikasi android ketika status keseluruhan “waspada”.



Gambar 3. 8 Contoh rancangan antarmuka aplikasi android ketika status keseluruhan “bahaya”.

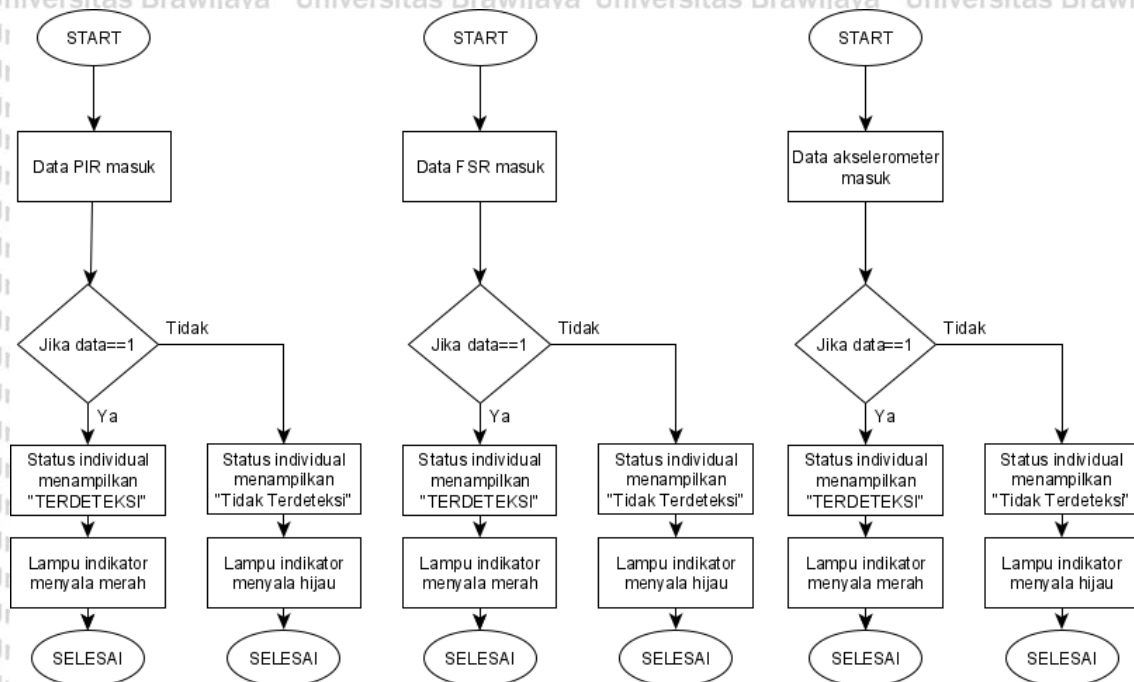
3.4.2 Rancangan Proses Kerja Aplikasi Android

Berikut merupakan blok diagram proses kerja aplikasi android secara keseluruhan:



Gambar 3. 9 Blok diagram proses kerja aplikasi

Dari blok diagram di atas, proses yang terjadi pada sistem adalah sebagai berikut: Ketika aplikasi dibuka, maka aplikasi akan langsung meminta data dari kanal ThingSpeak. Data yang dimaksud adalah data sensor PIR, FSR, dan akselerometer. Data tersebut kemudian diproses aplikasi dan ditampilkan dalam bentuk lampu indikator dan status individual. Pemrosesan data tersebut dijabarkan sebagai berikut:



Gambar 3. 10 Diagram alir pemrosesan data yang masuk ke aplikasi

- Data PIR, FSR, dan akselerometer akan masuk pada saat yang bersamaan.
- Jika data PIR terbaca “1”, maka status individual untuk “Kunci” akan menampilkan kata “TERDETEKSI” dan lampu indikator “Kunci” akan menyala merah.
- Jika data FSR terbaca “1”, maka status individual untuk “Jok” akan menampilkan kata “TERDETEKSI” dan lampu indikator “Jok” akan menyala merah.
- Jika data Akselerometer terbaca “1”, maka status individual untuk “Perpindahan” akan menampilkan kata “TERDETEKSI” dan lampu indikator “Perpindahan” akan menyala merah.
- Jika data PIR terbaca “0”, maka status individual untuk “Kunci” akan menampilkan kata “Tidak Terdeteksi” dan lampu indikator “Kunci” akan menyala hijau
- Jika data FSR terbaca “0”, maka status individual untuk “Jok” akan menampilkan kata “Tidak Terdeteksi” dan lampu indikator “Jok” akan menyala hijau.

g. Jika data Akselerometer terbaca “0”, maka status individual untuk “Perpindahan” akan menampilkan kata “Tidak Terdeteksi” dan lampu indikator “Perpindahan” akan menyala hijau.

Selanjutnya, kombinasi spesifik dari status individual akan membentuk status keseluruhan tertentu. Kombinasi status individual yang membentuk status keseluruhan dapat dilihat pada Tabel 3.1.

Tabel 3.1 Kombinasi status individual yang membentuk status keseluruhan

Status Keseluruhan	Status Kunci	Status Jok	Status Perpindahan	Keterangan
Bahaya	Terdeteksi	Terdeteksi	Terdeteksi	Paling berbahaya, kunci dijemol, motor diduduki dan dibawa pergi
Bahaya	Tidak Terdeteksi	Terdeteksi	Terdeteksi	Paling berbahaya, motor diduduki dan dibawa pergi
Bahaya	Terdeteksi	Terdeteksi	Tidak Terdeteksi	Berbahaya, kunci dijemol dan jok diduduki
Bahaya	Terdeteksi	Tidak Terdeteksi	Terdeteksi	Berbahaya, kunci dijemol dan motor sudah digerakkan tapi belum dibawa pergi
Bahaya	Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Berbahaya, kunci dijemol namun belum ada tindakan
Waspada	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Motor bergerak namun kunci tidak dijemol, bisa jadi dipindahkan tukang parkir, status waspada
Waspada	Tidak Terdeteksi	Terdeteksi	Tidak Terdeteksi	Motor diduduki namun tidak bergerak, bisa

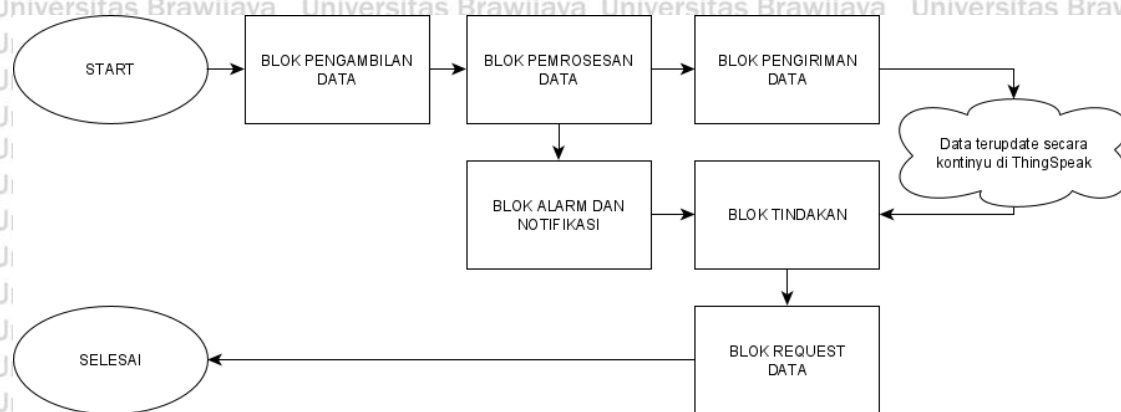
				jadi bukan pencurian, status waspada
Aman	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Motor kondisi aman

Jika status keseluruhan menunjukkan kata “WASPADA” atau “AMAN”, maka alarm akan menyala otomatis, yang artinya tombol “NYALAKAN ALARM” akan berubah menjadi tombol “MATIKAN ALARM”. Selanjutnya ada tiga tindakan yang bisa dilakukan oleh pengguna:

1. Jika pengguna mengetuk tombol “refresh”, maka aplikasi akan mengulang proses dari meminta data ke kanal ThingSpeak hingga menampilkan status keseluruhan keamanan motor.
2. Jika pengguna mengetuk tombol “LACAK”, maka aplikasi akan membuka layar baru dan menampilkan beberapa titik lokasi terakhir yang dilalui oleh motor menggunakan website Google Maps yang ditampilkan dalam aplikasi. Kemudian jika pengguna mengetuk tombol “kembali” pada layar tersebut, aplikasi akan kembali membuka halaman utama dan mengulangi proses dari meminta data ke kanal ThingSpeak hingga menampilkan status keseluruhan keamanan motor.
3. Jika pengguna mengetuk tombol “MATIKAN ALARM”, maka aplikasi akan mengirimkan status alarm “mati” ke ThingSpeak, kemudian Arduino yang membaca data tersebut dari ThingSpeak akan mematikan alarm.

3.5 Proses Kerja Sistem

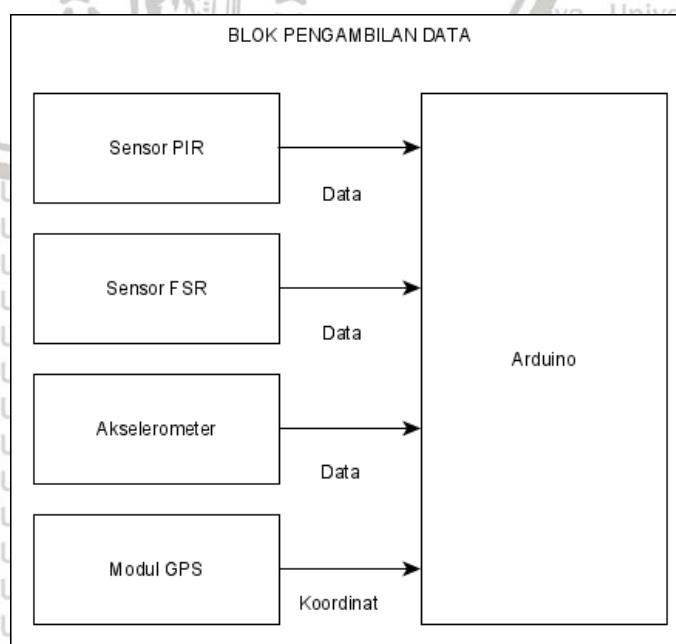
Pada gambar 3.11 dapat dilihat blok diagram proses kerja sistem secara keseluruhan.



Gambar 3.11 Blok diagram proses kerja sistem

Dari blok diagram pada gambar 3.11, proses yang terjadi pada sistem adalah sebagai berikut: Pertama-tama, setelah sistem dinyalakan melalui aplikasi, eksekusi blok pengambilan data. Setelah itu, data diproses dalam blok pemrosesan data. Selanjutnya, data dikirimkan ke ThingSpeak pada blok pengiriman data. Data kemudian terupdate secara kontinyu di kanal ThingSpeak. Pada saat yang bersamaan dengan pengiriman data, jika dari blok pemrosesan data terbukti bahwa terjadi gejala pencurian motor, maka aktifkan blok alarm dan notifikasi. Setelah itu, pengguna dapat melakukan beberapa aksi di dalam blok tindakan. Setelah terjadi tindakan, terjadi permintaan data dari ThingSpeak pada blok request data. Proses akan berulang secara terus-menerus hingga pengguna mematikan sistem melalui aplikasi, yang akan mengakhiri proses kerja sistem. Penjelasan mengenai isi dari blok-blok pada diagram blok keseluruhan sistem akan dijelaskan pada bagian selanjutnya.

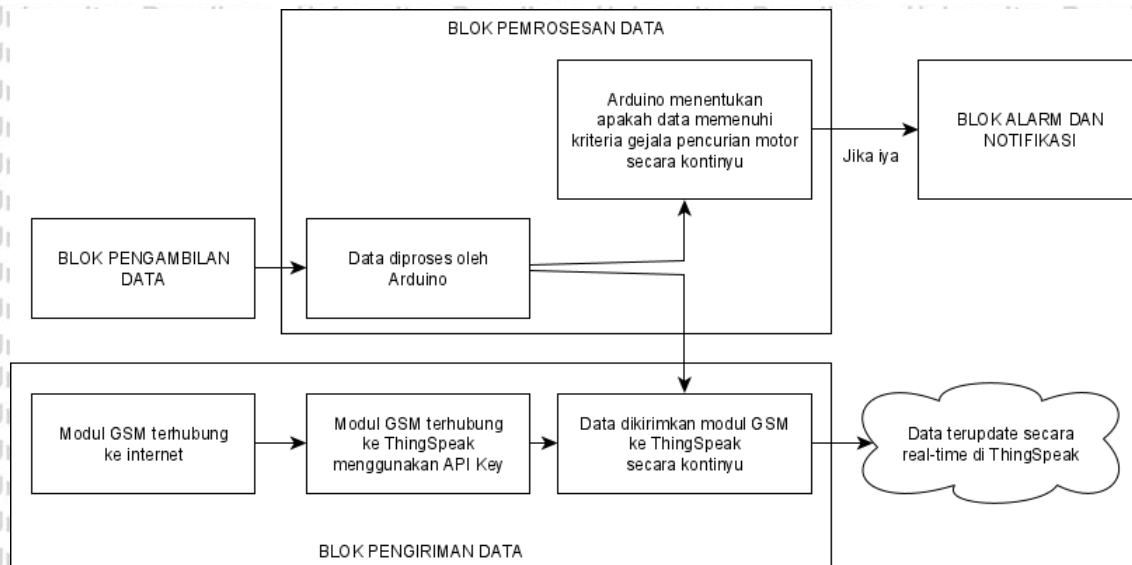
a. Blok pengambilan data



Gambar 3.12 Blok pengambilan data

Pada blok ini, data diambil dari sensor PIR, sensor FSR, dan akselerometer. Modul GPS juga mengirimkan data berupa koordinat, yang merupakan hasil pemilahan menggunakan library. Semua data dan koordinat diambil pada saat yang bersamaan dan dikirimkan ke Arduino segera setelah data diambil.

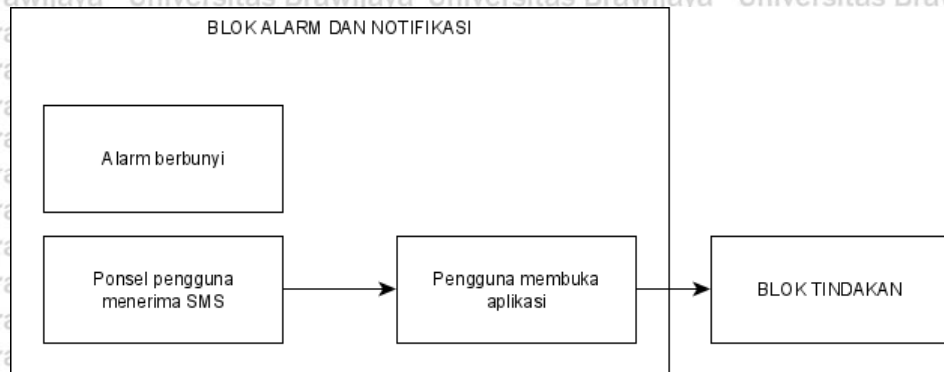
b. Blok pemrosesan dan pengiriman data



Gambar 3.13 Blok pemrosesan dan pengiriman data

Pada blok pemrosesan dan pengiriman data, ada beberapa hal yang terjadi pada saat yang bersamaan. Pada saat blok pengambilan data sedang dieksekusi, modul GSM mencoba terhubung ke internet. Kemudian, data diproses oleh Arduino. Pada saat ini modul GSM telah terhubung ke ThingSpeak menggunakan API Key. Setelah data diproses, data dikirimkan oleh modul GSM ke kanal ThingSpeak secara kontinyu, sehingga data pada ThingSpeak akan selalu terbaru setiap waktu. Pada saat yang bersamaan dengan pengiriman data ke ThingSpeak, Arduino secara kontinyu selalu membandingkan data yang masuk dengan data yang telah diset sebagai parameter gejala pencurian motor. Seperti yang telah disebutkan sebelumnya, parameter gejala pencurian motor adalah: di depan kunci motor ada aktifitas, atau adanya tekanan pada jok motor, atau adanya pergerakan motor. Jika salah satu dari ketiga parameter ini terpenuhi, maka sistem akan mengaktifkan blok berikutnya yaitu blok alarm dan notifikasi.

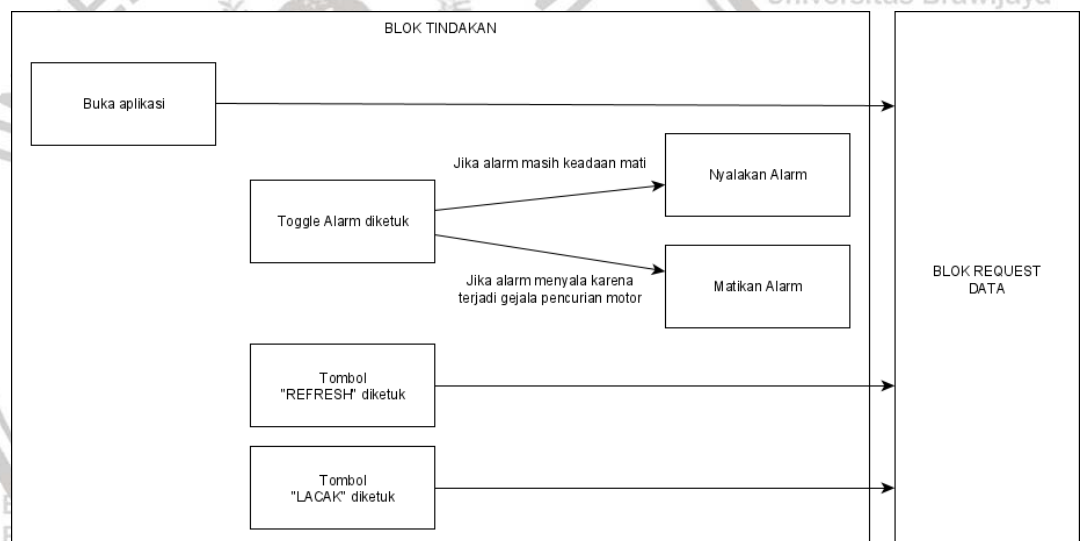
c. Blok alarm dan notifikasi



Gambar 3.14 Blok alarm dan notifikasi

Pada blok ini, alarm berbunyi pada jok motor dan pada saat yang bersamaan sebuah SMS akan terkirim ke ponsel pengguna. Pengguna dapat membuka aplikasi untuk mengetahui status motor. Selanjutnya pengguna dapat memilih untuk melakukan beberapa tindakan, yang akan dijelaskan pada blok tindakan

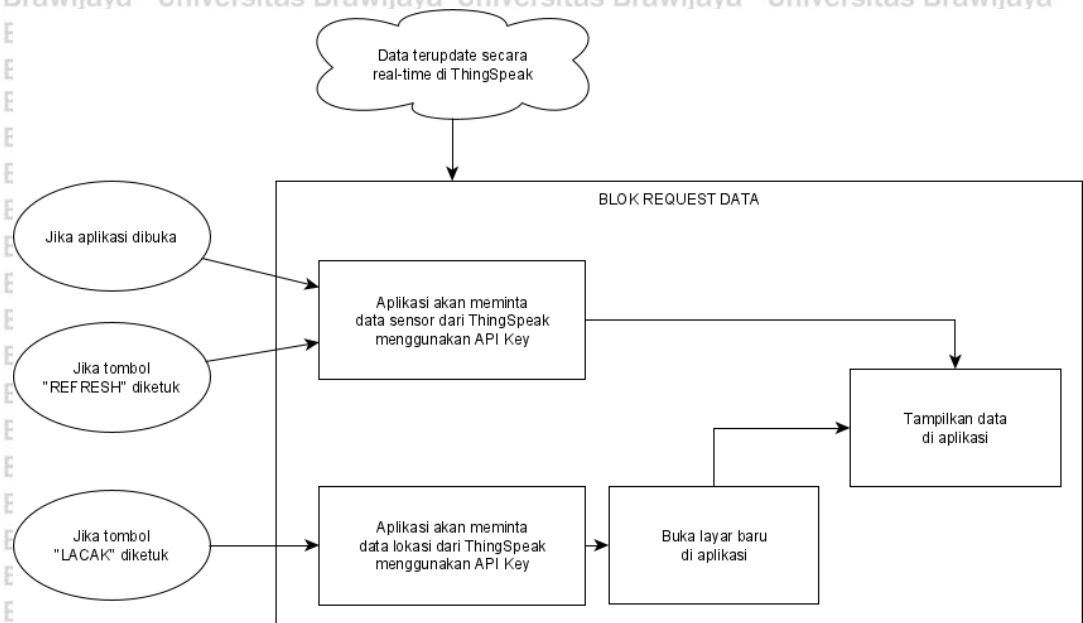
d. Blok tindakan



Gambar 3.15 Blok tindakan

Blok request data akan teraktivasi jika pengguna membuka aplikasi. Pengguna memiliki tiga buah pilihan: menyalakan atau mematikan alarm dengan mengetuk toggle alarm, memperbarui data yang ditampilkan di aplikasi dengan mengetuk tombol “refresh”, dan meminta lokasi motor dengan mengetuk tombol “lacak.” Jika pengguna mengetuk toggle alarm maka alarm akan nyala atau mati tergantung keadaan alarm sebelum toggle diketuk. Dan jika tombol “refresh” atau “lacak” diketuk maka sistem akan melanjutkan ke blok request data.

e. Blok request data



Gambar 3.16 Blok request data

Pada blok ini dapat dilihat bahwa input yang berbeda akan menghasilkan proses yang sedikit berbeda pada aplikasi. Jika tindakan yang dilakukan pengguna adalah membuka aplikasi atau mengetuk tombol “refresh”, maka data yang diminta aplikasi dari ThingSpeak adalah data dari ketiga sensor yang akan segera ditampilkan di layar aplikasi. Namun apabila pengguna mengetuk tombol “lacak”, maka data yang diminta oleh aplikasi dari ThingSpeak adalah data koordinat motor dari modul GPS. Untuk menampilkannya, aplikasi akan membuka layar baru, baru kemudian menampilkan data koordinat tersebut. Data yang akan diminta oleh aplikasi merupakan data terbaru yang selalu terbaru secara kontinyu di kanal ThingSpeak. Untuk memintanya, pada listing blok pemrograman aplikasi, akan diberikan Read API Key sebagai kunci untuk mengakses kanal ThingSpeak.

3.6 Pengujian Sistem

Setelah sistem sistem deteksi dini pencurian motor berbasis IoT dengan Arduino dan modul GSM dibuat secara keseluruhan, maka langkah selanjutnya adalah melakukan pengujian dan analisis terhadap keseluruhan sistem. Pengujian yang dilakukan meliputi:

1. Pengujian performa sensor PIR.

Pengujian ini bertujuan untuk memastikan bahwa sensor PIR yang tersemat pada sistem dapat mendeteksi adanya gejala pencurian motor; yaitu adanya gerakan manusia di depan lubang kunci motor, dalam kondisi cuaca cerah maupun hujan, dan pada waktu pagi, siang, maupun malam. Pengujian dilakukan pada waktu dan cuaca yang berbeda-beda karena sensor PIR sensitif terhadap cahaya dan suhu lingkungan.

2. Pengujian performa sensor FSR.

Pengujian ini bertujuan untuk memastikan bahwa sensor FSR yang tersemat pada sistem dapat mendeteksi adanya gejala pencurian motor; yaitu adanya tekanan pada jok motor pada waktu pagi, sore, maupun malam. Pengujian dilakukan pada waktu yang berbeda-beda karena suhu jok motor mempengaruhi tekanan yang akan diterima oleh sensor FSR.

3. Pengujian performa akselerometer.

Pengujian ini bertujuan memastikan bahwa akselerometer yang tersemat pada sistem dapat mendeteksi adanya gejala pencurian motor; yaitu adanya perpindahan motor dari suatu tempat ke tempat lain.

4. Pengujian akurasi modul GPS.

Pengujian ini bertujuan untuk mengukur tingkat keakurasian modul GPS.

5. Pengujian sistem komunikasi data secara keseluruhan.

Pengujian ini bertujuan untuk memastikan bahwa modul GSM mampu mengirimkan data dari sistem ke ThingSpeak yang kemudian ditampilkan pada aplikasi dan sebaliknya mampu menerima data dari ThingSpeak dan memprosesnya pada sistem.

6. Pengujian sistem secara keseluruhan.

Pengujian ini bertujuan untuk memastikan bahwa sistem bekerja sesuai dengan harapan; yaitu mampu memenuhi spesifikasi sistem.

3.7 Pengambilan Kesimpulan dan Saran

Kesimpulan didapat berdasarkan hasil penerapan dan pengujian sesuai dengan tujuan dan rumusan masalah yang ditentukan. Sehingga, dapat dikatakan bahwa apabila hasil yang didapatkan telah sesuai dengan yang direncanakan sebelumnya, maka penerapan sistem deteksi dini pencurian motor berbasis IoT dengan Arduino dan modul GSM tersebut telah berhasil mencapai target. Saran diberikan setelah melihat bahwa masih terdapat kekurangan dalam sistem yang dibuat dengan harapan adanya pengembangan dan perbaikan terhadap realisasi sistem agar didapatkan hasil yang lebih baik.

BAB IV HASIL DAN PEMBAHASAN

Hasil-hasil yang tertera pada penelitian ini diambil dari pengujian sistem. Pengujian sistem dilakukan untuk mengetahui apakah sistem dapat bekerja sesuai dengan harapan.

Pengujian-pengujian yang dilakukan pada penelitian ini meliputi:

1. Pengujian Performa Sensor PIR
2. Pengujian Performa Sensor FSR
3. Pengujian Performa Akselerometer
4. Pengujian Akurasi Modul GPS
5. Pengujian Sistem Komunikasi Data Secara Keseluruhan
6. Pengujian Sistem Secara Keseluruhan

4.1 Pengujian Performa Sensor PIR

Pengujian ini bertujuan untuk memastikan bahwa sensor PIR yang tersemat pada sistem dapat mendeteksi adanya gejala pencurian motor; yaitu adanya gerakan manusia di depan lubang kunci motor, dalam kondisi cuaca cerah maupun hujan, dan pada waktu pagi, siang, maupun malam. Pengujian dilakukan pada waktu dan cuaca yang berbeda-beda karena sensor PIR sensitif terhadap cahaya dan suhu lingkungan.

4.1.1 Prosedur Pengujian

- Menyematkan sensor PIR di bawah jok motor pada area kaki pengemudi.
- Untuk menghindari pendeteksian gerakan di samping kiri maupun kanan motor, sensor PIR ditutupi dengan karton buffalo hitam pada sisi kiri, kanan, dan bawah lensa Fresnel PIR.
- Menggerakkan tangan di area lubang kunci motor.
- Memastikan apakah sensor PIR mendeteksi gerakan di area lubang kunci motor melalui hasil yang tertera di serial monitor.
- Melakukan pengujian dengan kondisi waktu pagi, sore, serta malam hari pada saat cuaca cerah maupun cuaca hujan yang disimulasikan.
- Cuaca hujan disimulasikan dengan menyemprot permukaan lensa Fresnel PIR dengan air secara menyeluruh.
- Mencatat hasil pengujian pada semua kondisi.

Dalam pengujian ini pengaturan sensitivitas PIR diatur menjadi 3 meter, dan delaynya diatur menjadi 5 detik. Mode yang digunakan adalah mode *repeatable*. Tempat yang dipilih untuk pengujian ini yaitu area di depan lokasi rumah penulis. Pengujian dilakukan dengan menghubungkan sensor PIR ke Arduino yang terhubung ke laptop.

4.1.2 Hasil Pengujian dan Analisis

Sensor PIR dinyatakan berhasil mendeteksi adanya gejala pencurian motor apabila sensor PIR dapat mendeteksi gerakan di area lubang kunci motor pada segala kondisi. Hasil pengujian performa sensor PIR dirangkum pada Tabel 4.1.

Tabel 4. 1 Hasil pengujian performa sensor PIR

Waktu	Cuaca	Terdeteksi	
		Tanpa Input	Dengan Input
Pagi (Pukul 10.00-10.15 WIB)	Cerah	Tidak	Ya
	Hujan	Ya	Ya
Sore (Pukul 15.00-15.15 WIB)	Cerah	Tidak	Ya
	Hujan	Ya	Ya
Malam (Pukul 20.00-20.15 WIB)	Cerah	Tidak	Ya
	Hujan	Tidak	Ya

Berdasarkan hasil pengujian, sensor PIR mampu mendeteksi gerakan di area lubang kunci motor pada waktu pagi, sore, dan malam pada saat kondisi cuaca cerah. Hasil pembacaan mulai tidak konsisten pada saat kondisi hujan di waktu pagi dan sore. Hal ini dikarenakan lensa Fresnel terhalang oleh titik-titik air hujan, sedangkan air dapat menyerap radiasi sinar inframerah. Jika titik-titik air hujan tersebut terekspos ke panas tubuh manusia atau binatang sebelum menyentuh lensa Fresnel, sensor PIR akan menyalahartikan titik-titik hujan tersebut sebagai input gerakan.

4.2 Pengujian Performa Sensor FSR

Pengujian ini bertujuan untuk memastikan bahwa sensor FSR yang tersemat pada sistem dapat mendeteksi adanya gejala pencurian motor; yaitu adanya tekanan pada jok motor pada waktu pagi, sore, maupun malam. Pengujian dilakukan pada waktu yang berbeda-beda karena suhu jok motor mempengaruhi tekanan yang akan diterima oleh sensor FSR.

4.2.1 Prosedur Pengujian

- Sensor FSR dimasukkan ke dalam jok motor.
- Dilakukan pengujian dengan menekan jok motor selama 1 detik sampai terbaca hasil pembacaan di serial monitor sebesar 10 sampai 500.
- Dilakukan pengujian dengan menduduki jok motor selama 2 menit sampai terbaca hasil pembacaan di serial monitor sebesar lebih dari 500.
- Mencatat hasil yang tertera pada serial monitor.
- Mengulangi pengujian pada waktu pagi, sore, maupun malam.

Dalam pengujian ini yang menekan dan menduduki jok motor adalah penulis dengan berat badan sekitar 65 kg. Tempat yang dipilih untuk pengujian ini yaitu area di depan lokasi rumah penulis. Pengujian dilakukan dengan menghubungkan sensor FSR ke Arduino yang terhubung ke laptop.

4.2.2 Hasil Pengujian dan Analisis

Sensor FSR dinyatakan berhasil mendeteksi adanya gejala pencurian motor ketika sensor dapat mendeteksi adanya tekanan tekanan yang konstan pada jok motor. Hasil pengujian performa sensor FSR dirangkum pada Tabel 4.2.

Tabel 4. 2 Hasil pengujian performa sensor FSR

Waktu	Tindakan dan Lama Tekanan	Hasil Pembacaan Sensor	Terdeteksi	
			Tanpa Input	Dengan Input
Pagi (Pukul 10.00-10.15 WIB)	Ditekan dengan tangan, 2 detik	143-236	Tidak	Ya
	Diduduki, 2 menit	960-985	Tidak	Ya
Sore (Pukul 15.00-15.15 WIB)	Ditekan dengan tangan, 2 detik	254-481	Tidak	Ya
	Diduduki, 2 menit	802-984	Tidak	Ya
Malam (Pukul 20.00-20.15 WIB)	Ditekan dengan tangan, 2 detik	322-479	Tidak	Ya
	Diduduki, 2 menit	898-999	Tidak	Ya

Berdasarkan data hasil pengujian, sensor FSR mampu mendeteksi tekanan konstan dalam segala kondisi. Hasil pembacaan sensor berbeda-beda pada waktu pagi, siang, dan sore. Untuk percobaan dengan tindakan menekan jok motor dengan tangan, hasil pembacaan sangat bervariasi. Hal ini dikarenakan tekanan tangan pada setiap pengujian tidak konsisten

karena sebesar apa tekanan tangan itu sendiri sulit diukur. Namun, hal ini bukanlah sebuah masalah, karena ambang pendeteksian gejala pencurian motor pada sensor FSR diatur pada angka 500, sedangkan hasil pembacaan sensor FSR ketika ditekan dengan tangan semuanya masih berada di bawah angka 500. Untuk percobaan dengan tindakan menduduki jok motor, perbedaan hasil pembacaan tidak signifikan di antara ketiga waktu yang berbeda, yang artinya sensor FSR mampu mendeteksi gejala pencurian motor secara konsisten pada segala kondisi. Penting untuk dicatat bahwa pada saat pengujian dilakukan, ketika jok motor tidak diberikan tekanan dalam bentuk apapun, sensor FSR tidak mendeteksi tekanan apapun pada waktu pagi, siang, maupun malam.

4.3 Pengujian Performa Akselerometer

Pengujian ini bertujuan untuk memastikan bahwa akselerometer yang tersemat pada sistem dapat mendeteksi adanya gejala pencurian motor; yaitu adanya perpindahan motor dari suatu tempat ke tempat lain.

4.3.1 Prosedur Pengujian

- Meletakkan akselerometer di atas jok motor.
- Memindahkan motor dengan mendorongnya maju dari penanda pertama ke penanda ke dua sejauh satu meter kemudian berhenti.
- Memeriksa apakah akselerometer mampu mendeteksi adanya perpindahan melalui hasil yang tertera pada serial monitor.
- Mengulangi prosedur sebelumnya dari penanda ke dua ke penanda ke tiga. Ulangi terus sampai motor sampai ke penanda ke 6 pada meter ke lima.
- Memindahkan motor dengan menariknya mundur dari penanda ke enam ke penanda ke lima sejauh satu meter kemudian berhenti.
- Memeriksa apakah akselerometer mampu mendeteksi adanya perpindahan melalui hasil yang tertera pada serial monitor.
- Mengulangi prosedur sebelumnya dari penanda ke lima ke penanda ke empat. Ulangi terus sampai motor sampai ke posisi awal penanda pertama pada meter ke nol.
- Mencatat hasil pengujian pada semua iterasi perpindahan.

Pengujian dilakukan di jalanan di samping rumah penulis dengan menggunakan garis penanda di aspal yang dibuat dengan merentangkan tali sejauh 5 meter, kemudian menandai jarak setiap 1 meter menggunakan cat semprot. Motor digerakkan dengan mendorongnya maju dari penanda satu ke penanda berikutnya kemudian menariknya mundur dari penanda

ke enam ke penanda ke lima dan seterusnya. Kecepatan penggerakan motor kurang dari 10km/jam. Pengujian dilakukan dengan menghubungkan akselerometer ke Arduino yang terhubung ke laptop.

4.3.2 Hasil Pengujian dan Analisis

Akselerometer dinyatakan berhasil apabila akselerometer dapat mendeteksi semua perpindahan pada semua iterasi perpindahan. Hasil pengujian performa akselerometer dirangkum pada Tabel 4.3.

Tabel 4.3 Hasil pengujian performa akselerometer

Perpindahan yang dilakukan	Terdeteksi
Maju pertama	Ya
Maju ke dua	Ya
Maju ke tiga	Ya
Maju ke empat	Ya
Maju ke lima	Ya
Mundur pertama	Ya
Mundur ke dua	Ya
Mundur ke tiga	Ya
Mundur ke empat	Ya
Mundur ke lima	Ya

Berdasarkan hasil pengujian, akselerometer mampu mendeteksi adanya pergerakan motor selama pergerakan tersebut terjadi pada semua perpindahan yang dilakukan. Akselerometer juga akan mendeteksi pergerakan apabila terjadi gerakan yang tiba-tiba, sebagai contoh ketika motor bergoyang, ketika motor dimiringkan, dan ketika motor direm.

4.4 Pengujian Akurasi Modul GPS

Pengujian ini bertujuan untuk mengukur tingkat keakurasian modul GPS. Pengujian dilakukan dengan memindahkan modul GPS yang tersemat pada Arduino kemudian melihat posisi bujur lintang yang terbaca oleh modul GPS dibandingkan dengan posisi bujur lintang

sebenarnya yang terbaca oleh ponsel pintar, lalu melihat jarak antara keduanya menggunakan aplikasi Google Maps.

4.4.1 Prosedur Pengujian

- Pengujian dilakukan dengan membawa modul GPS ke luar ruangan.
- Meletakkan modul GPS pada titik awal pengujian, yaitu teras rumah penulis.
- Mencatat hasil pembacaan modul GPS pada serial monitor.
- Mencatat posisi modul GPS sebenarnya menggunakan website mylocation.org yang diakses menggunakan ponsel pintar.
- Mengukur selisih jarak titik lokasi sebenarnya dengan titik lokasi yang terbaca oleh modul GPS menggunakan aplikasi Google Maps pada ponsel pintar yang berada pada lokasi yang sama.
- Memindahkan modul GPS ke lokasi di sekitar rumah penulis yang dipilih secara acak.
- Mengulangi prosedur sebelumnya sampai tercatat lima lokasi.

Pengujian dilakukan dengan berpindah dari posisi satu ke posisi berikutnya menggunakan sepeda motor. Modul GPS disematkan ke Arduino yang terhubung dengan laptop untuk melihat hasil pembacaan melalui serial monitor.

4.4.2 Hasil Pengujian dan Analisis

Keakurasian modul GPS dapat diukur dengan membandingkan hasil pembacaan bujur lintang oleh modul GPS dengan bujur lintang sebenarnya yang didapatkan dari website mylocation.org dan melihat selisih jarak dari titik lokasi sebenarnya ke titik lokasi yang dibaca oleh modul GPS menggunakan aplikasi Google Maps. Hasil pengujian keakurasian modul GPS dapat dilihat pada Tabel 4.4.

Tabel 4. 4 Hasil pengujian keakurasian modul GPS

Lokasi	Hasil Pembacaan Bujur Lintang oleh Modul GPS	Bujur Lintang Sebenarnya dari mylocation.org	Selisih jarak antara Kedua Titik Bujur Lintang
1	-6.265425,106.989952	-6.265474,106.989666	20m
2	-6.264445,106.988975	-6.264450,106.988955	0m
3	-6.264427,106.990982	-6.264436,106.990943	10m

4	-6.263434,106.991615	-6.263456,106.991619	0m
5	-6.262919,106.990066	-6.263068,106.990041	10m

Berdasarkan hasil pengujian, dapat dilihat bahwa dari lima lokasi pengujian, terdapat dua lokasi di mana ketelitian modul GPS sangat akurat, di mana selisih jarak antara kedua bujur lintang yang terekam adalah 0 meter. Terdapat juga dua lokasi lain di mana terdapat perbedaan jarak tempuh sebesar 10 meter. Selisih jarak yang paling jauh di antara kedua titik bujur lintang yang diuji adalah 20 meter. Dari sini dapat disimpulkan bahwa tingkat ketidakakurasian modul GPS paling tinggi adalah sejauh 20 meter dari lokasi sesungguhnya, dan modul GPS dapat mendeteksi paling akurat hingga 0 meter dari lokasi sesungguhnya.

4.5 Pengujian Sistem Komunikasi Data Secara Keseluruhan

Pengujian ini bertujuan untuk memastikan bahwa modul GSM mampu mengirimkan data dari sistem ke ThingSpeak yang kemudian ditampilkan pada aplikasi dan sebaliknya mampu menerima data dari ThingSpeak dan memprosesnya pada sistem.

4.5.1 Prosedur Pengujian

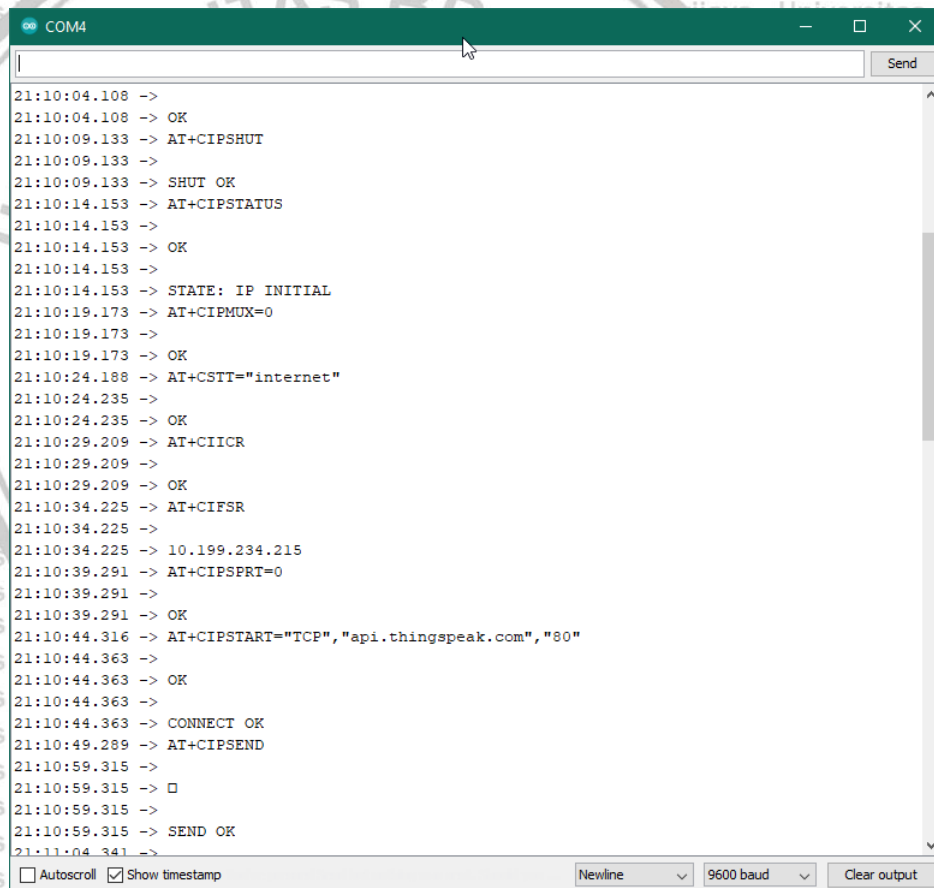
- Memasang sistem pada motor.
- Menyalakan sistem dengan menghubungkan Arduino dengan sumber daya.
- Memastikan bahwa modul GSM sudah tersambung ke Internet.
- Sistem mengirimkan data ke ThingSpeak.
- Memeriksa data yang terbaru di ThingSpeak.
- Mencatat jeda waktu antara pengiriman data dari sistem dengan munculnya data pada ThingSpeak.
- Menyalakan alarm dari aplikasi.
- Mencatat jeda waktu antara aktivasi alarm dari aplikasi dengan aktifnya alarm pada sistem.
- Mematikan alarm dari aplikasi.
- Mencatat jeda waktu antara penonaktifan alarm dari aplikasi dengan matinya alarm pada sistem.
- Mematikan sistem dari aplikasi.
- Memastikan kenonaktifan sistem dengan cara memberi input pada salah satu sensor.

- Mencatat jeda waktu antara penonaktifan sistem dari aplikasi dengan kenonaktifan sistem secara riil.
- Menyalakan sistem dari aplikasi.
- Memastikan keaktifan sistem dengan cara memberi input pada salah satu sensor.
- Mencatat jeda waktu antara aktivasi sistem dari aplikasi dengan aktifnya sistem secara riil.

Pengujian dilakukan dengan menggunakan aplikasi yang sudah terpasang pada ponsel pintar.

4.5.2 Hasil Pengujian dan Analisis

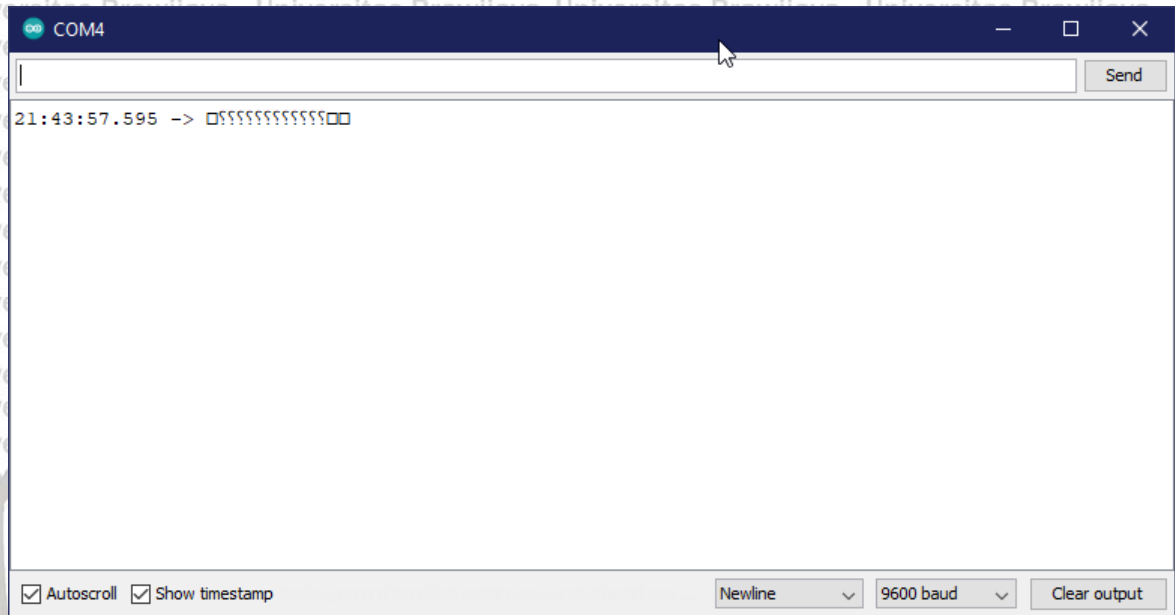
Berdasarkan hasil pengujian, terdapat kendala pada alur kerja sistem pada bagian pemrosesan data menuju pengiriman data, yang dapat dilihat pada Gambar 4.1.



Gambar 4. 1 Tampilan serial monitor pada saat fungsi pengiriman data dijalankan

Pada Gambar 4.1 bisa dilihat bahwa setelah modul GSM terhubung dengan ThingSpeak, data yang mau dikirimkan yang bertipe string dan berupa gabungan dari API key dan semua data masukan dari sistem tidak terproses dengan benar, sehingga hanya

tampil berupa karakter tidak dikenal. Data string tersebut akhirnya tidak terkirim dengan sukses ke ThingSpeak. Selain itu, apabila fungsi pengiriman data dan permintaan data ke ThingSpeak diaktifkan pada sistem, terjadi kelebihan memori pada Arduino yang mengakibatkan sistem menjadi tidak stabil dan data tidak dapat ditampilkan pada serial monitor.



Gambar 4. 2 Tampilan serial monitor ketika keseluruhan fungsi pengiriman dan permintaan data dijalankan pada sistem.

Dari sini dapat ditarik kesimpulan bahwa sistem komunikasi data tidak dapat bekerja sesuai dengan harapan.

4.6 Pengujian Sistem Secara Keseluruhan

Pengujian ini bertujuan untuk memastikan bahwa sistem bekerja sesuai dengan harapan; yaitu mampu memenuhi spesifikasi sistem yang telah diuraikan pada Bab III.

4.6.1 Prosedur Pengujian

- Memasang sistem pada motor.
- Memastikan bahwa modul GSM sudah tersambung ke Internet.
- Memastikan bahwa modul GPS sudah mendapatkan sinyal.
- Menguji satu per satu kondisi yang tertera pada tabel berikut dengan status yang muncul pada aplikasi.

Tabel 4. 5 Kondisi pada setiap sensor dan status pada aplikasi

Status Kunci	Status Jok	Status Perpindahan	Status Keseluruhan
Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Aman
Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Waspada
Tidak Terdeteksi	Terdeteksi	Tidak Terdeteksi	Waspada
Tidak Terdeteksi	Terdeteksi	Terdeteksi	Bahaya
Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Bahaya
Terdeteksi	Tidak Terdeteksi	Terdeteksi	Bahaya
Terdeteksi	Terdeteksi	Tidak Terdeteksi	Bahaya
Terdeteksi	Terdeteksi	Terdeteksi	Bahaya

4.6.2 Hasil Pengujian dan Analisis

Sistem dapat dikatakan berhasil apabila semua hasil yang ditampilkan pada aplikasi sesuai dengan kondisi pada Tabel 4.5. Berdasarkan hasil pengujian, dikarenakan adanya kegagalan pemrosesan data pada sistem yang mengakibatkan data tidak dapat dikirim maupun diterima dari ThingSpeak, maka dapat dikatakan sistem secara keseluruhan tidak dapat bekerja sesuai dengan harapan.

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan pengujian dan analisis sistem deteksi dini gejala pencurian motor berbasis IoT dengan Arduino dan modul GSM, dapat ditarik beberapa kesimpulan sebagai berikut:

- 1) Sistem tidak dapat bekerja sesuai dengan harapan penulis.
- 2) Semua komponen dapat bekerja dengan baik apabila dites satu per satu, kecuali pembacaan dari sensor PIR yang kurang bisa diandalkan pada saat kondisi hujan di waktu pagi dan sore, namun hal tersebut tidak mempengaruhi kinerja pendeteksian gejala pencurian motor secara signifikan.
- 3) Sistem sangat bergantung pada kondisi sinyal GSM dan GPS.
- 4) Deviasi titik lokasi GPS pada sistem maksimal 20 meter dari posisi sebenarnya.
- 5) Rancangan sistem tidak dapat diterapkan pada Arduino Uno dikarenakan kurangnya memori yang tersedia.

5.2 Saran

Adapun saran penulis untuk pengembangan lebih lanjut adalah sebagai berikut:

- 1) Sensor PIR dapat diganti dengan sensor lain yang lebih bisa diandalkan dalam kondisi hujan.
- 2) Modul GSM SIM800L dapat diganti dengan modul GSM lain yang kekuatannya pendeteksian sinyalnya lebih tinggi.
- 3) Modul GPS NEO6M-V2 dapat diganti dengan modul GPS lain yang tingkat keakurasiannya lebih tinggi.
- 4) Arduino Uno sebaiknya diganti dengan Arduino tipe lainnya yang kapasitas memorinya lebih tinggi.
- 5) Dapat ditambahkan fungsi untuk menyalakan atau mematikan sistem melalui aplikasi.
- 6) Sumber daya untuk sistem dapat diganti dari menggunakan *powerbank* dan baterai eksternal menjadi menggunakan aki motor.

DAFTAR PUSTAKA

- Adafruit. (2012). *Force Sensitive Resistor (FSR)*. <https://learn.adafruit.com/force-sensitive-resistor-fsr/overview>. (diakses 18 Mei 2020).
- Adafruit. (2014). *How PIRs Work*. <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work>. (diakses 18 Mei 2020).
- Adafruit. (n.d.). *Square Force-Sensitive Resistor (FSR) - Interlink 406*. <https://www.adafruit.com/product/1075>. (diakses 18 Mei 2020).
- Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: a survey. *Computer Networks Journal*, Vol. 54 No.15 Oktober, 2787-2805.
- BPS Kota Malang. (2017). *Jumlah Kendaraan Bermotor Berdasarkan Plat Nomor di Kota Malang, 2015-2016*. <https://malangkota.bps.go.id/statistictable/2017/07/21/601/jumlah-kendaraan-bermotor-berdasarkan-plat-nomor-di-kota-malang-2015-2016.html>. (diakses 3 April 2020).
- BPS Subdirektorat Statistik Politik dan Keamanan. (2019). *Statistik Kriminal 2019*. Jakarta: Badan Pusat Statistik.
- Components 101. (2017). *Active Passive Buzzer*. <https://components101.com/buzzer-pinout-working-datasheet>. (diakses 18 Mei 2020).
- Components 101. (2017). *HC-SR501 PIR Sensor*. <https://components101.com/hc-sr501-pir-sensor>. (diakses 18 Mei 2020).
- Components 101. (2018). *NEO-6MV2 GPS Module*. <https://components101.com/modules/neo-6mv2-gps-module>. (diakses 18 Mei 2020).
- Electronics For U. (2019). *All You Wanted to Know About GSM Module and GPRS Module*. <https://www.electronicsforu.com/resources/gsm-module>. (diakses 18 Mei 2020).
- Jawa Pos Radar Malang. (2017). *Kota Malang Darurat Curanmor, 9 Bulan 625 Motor Hilang*. <https://radarmalang.id/kota-malang-darurat-curanmor-9-bulan-625-motor-hilang/>. (diakses 3 April 2020).
- Kadir, A. (2012). *Panduan Praktis Mempelajari Aplikasi Mikrokontroler dan Pemrogramannya menggunakan Arduino*. Yogyakarta: Andi.
- Kartha, V. (n.d.). *Interfacing MPU-6050 / GY-521 board with Arduino Uno*. <https://electrosome.com/interfacing-mpu-6050-gy-521-arduino-uno/>. (diakses 18 Mei 2020).
- Last Minute Engineers. (n.d.). *Send Receive SMS & Call with SIM800L GSM Module & Arduino*. <https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/>. (diakses 18 Mei 2020).

NASA. (2014). *Observing in Infrared*.

<https://earthobservatory.nasa.gov/features/FalseColor/page5.php>. (diakses 22 Juni 2021).

Ruth, E. (2013). Deskripsi Kualitas Layanan Jasa Akses Internet di Indonesia dari Sudut Pandang Penyelenggara. *Buletin Pos dan Telekomunikasi*, Vol.11 No.2 Juni, 137-146.

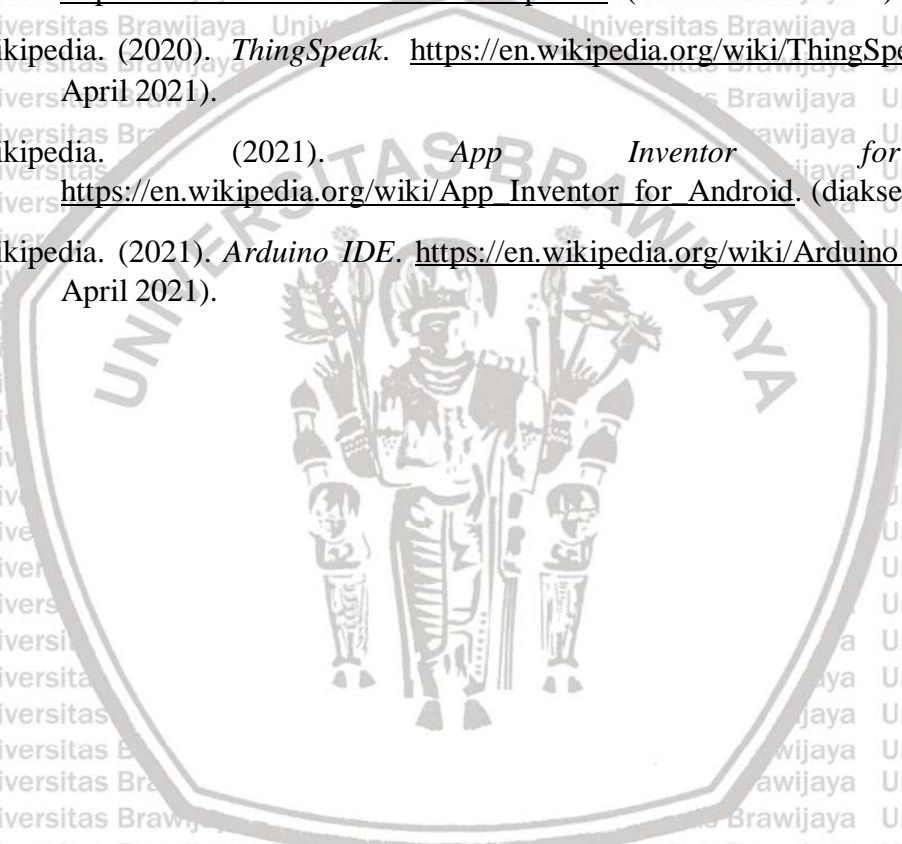
Santos, S. (2018). *Guide to NEO-6M GPS Module with Arduino*. <https://randomnerdtutorials.com/guide-to-neo-6m-gps-module-with-arduino/>. (diakses 18 Mei 2020).

Sulaiman, A. (2012). *Arduino: Mikrocontroller Bagi Pemula Hingga Mahir*. <http://buletin.balaielektronika.com/?p=163>. (diakses 2 Mei 2014).

Wikipedia. (2020). *ThingSpeak*. <https://en.wikipedia.org/wiki/ThingSpeak>. (diakses 27 April 2021).

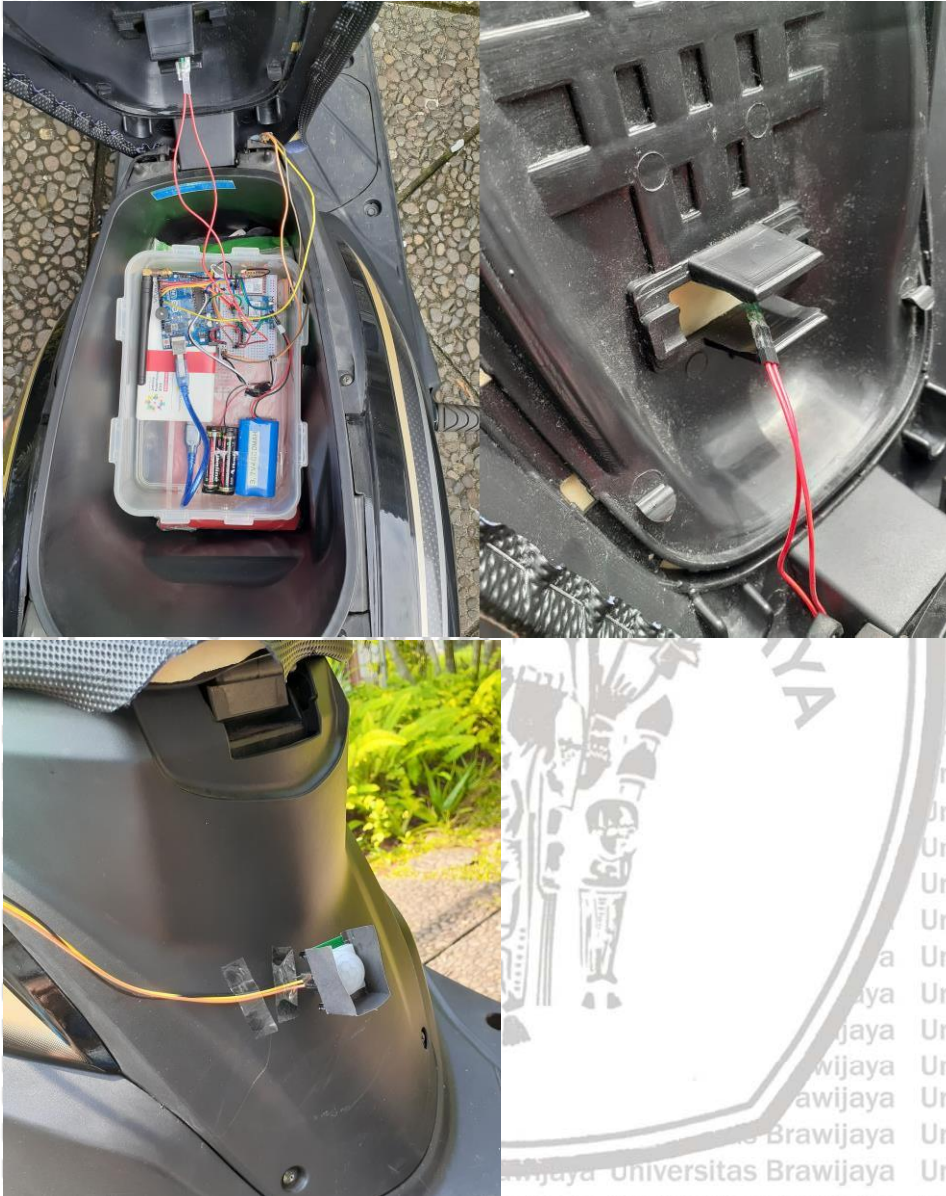
Wikipedia. (2021). *App Inventor for Android*. https://en.wikipedia.org/wiki/App_Inventor_for_Android. (diakses 27 April 2021)

Wikipedia. (2021). *Arduino IDE*. https://en.wikipedia.org/wiki/Arduino_IDE. (diakses 27 April 2021).

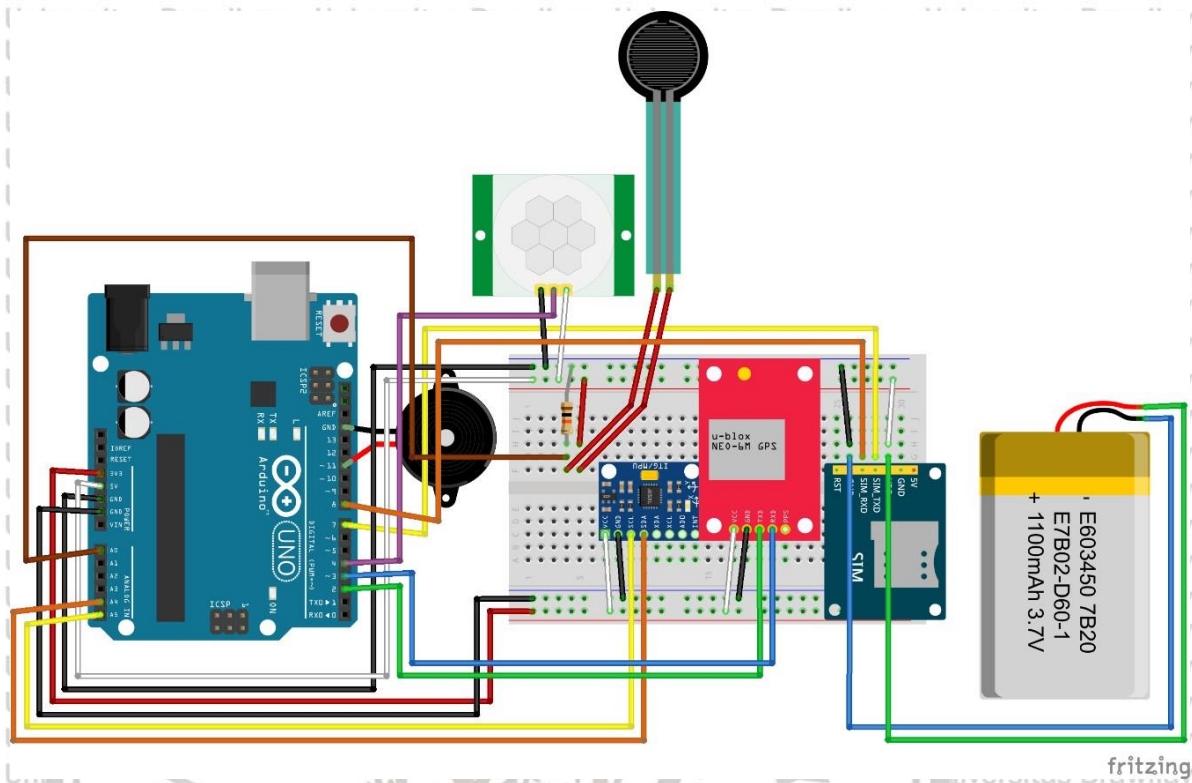


LAMPIRAN

Lampiran 1. Foto Alat



Lampiran 2. Skematik Breadboard Sistem



Lampiran 3. Sourcecode Arduino

```
#include "TinyGPS++.h"
#include <SoftwareSerial.h>
#include <String.h>
#include <Wire.h>
#include <MPU6050.h>

SoftwareSerial gprsSerial(7,8);
SoftwareSerial GPSSerial(2,3);
TinyGPSPlus gps;
MPU6050 mpu; //SCL A5 SDA A4

#define fsrPin A0
int pirSt = LOW;
int val = 0;
int fsrR;
String stPIR;
```

```
String stK;

String stJ;

String stP;

String datAMsk = "";

String stALst = "";

String datSMsk = "";

String stSLst = "";

String Lat;

String Lng;

void setup() {

  Serial.begin(9600);

  GPSSerial.begin(9600);

  gprsSerial.begin(9600);

  Wire.begin();

  pinMode(11, OUTPUT);//alarm

  pinMode(4, INPUT);//pir

  digitalWrite(11, LOW);//alarm

  digitalWrite(4, LOW);//pir

  stPIR="0";

  stK = "0";

  stJ = "0";

  stP = "0";

  stSLst="0";

  Lat = "-6.265436";

  Lng = "106.989580";

  datAMsk.reserve(200);

  datSMsk.reserve(200);

}
```



```

void loop() {
    MasukDataGPS();
    MasukDataPIR();
    MasukDataFSR();
    MasukDataAcc();
    SIM800LSEND();
    AlarmONOFF();
}

void MasukDataGPS(){
    while(GPSSerial.available()>0)  //Selama ada karakter masuk ke modul GPS
    {
        gps.encode(GPSSerial.read()); //Masukkan data NMEA ke library satu karakter per satu
        karakter
        if(gps.location.isUpdated())  //Untuk mengurangi kondisi supaya hanya dieksekusi ketika
        data NMEA selesai masuk
        {
            //Mengambil informasi terakhir dari objek gps yang berasal dari data yang diambil oleh
            modul GPS
            Serial.print("Bujur Lintang:");
            Lat = String(gps.location.lat(),6);
            Lng = String(gps.location.lng(),6);
            Serial.print(Lat); Serial.print(","); Serial.println(Lng);
            Serial.println("");
            delay(2000);
        }
    }
}

void ShowResponse(){
    while(gprsSerial.available() != 0)
    Serial.write(gprsSerial.read());
    delay(5000);
}

```

```

void GetResponse(int wait){
    datAMsk = "";
    Serial.print("response : ");
    long timeNOW2 = millis();
    while(millis()-timeNOW2 < wait){
        while(gprsSerial.available()>0){
            datAMsk += (char)gprsSerial.read();
        }
        delay(1);
    }
    Serial.println(datAMsk);
    Serial.println();
}

String parse(String data, char separator, int index)
{
    int found = 0;
    int strIndex[] = {0, -1};
    int maxIndex = data.length()-1;

    for(int i=0; i<=maxIndex && found<=index; i++){
        if(data.charAt(i)==separator || i==maxIndex){
            found++;
            strIndex[0] = strIndex[1]+1;
            strIndex[1] = (i == maxIndex) ? i+1 : i;
        }
    }

    return found>index ? data.substring(strIndex[0], strIndex[1]) : "";
}

```



```
void MasukDataPIR(){
    val = digitalRead(4);
    if (val == HIGH) {
        if (pirSt == LOW) {
            Serial.println("GERAKAN MANUSIA TERDETEKSI");
            stPIR = "1";
            stK = "1";
            pirSt = HIGH;
            Serial.print("Status Kunci = ");
            Serial.println(stK);
        }
    }
    else {
        if (pirSt == HIGH) {
            Serial.println("GERAKAN BERHENTI");
            stPIR = "0";
            pirSt = LOW;
            Serial.print("Status Kunci = ");
            Serial.println(stK);
        }
    }
}
```

```
void MasukDataFSR(){
    fsrR = analogRead(fsrPin);
    Serial.print("Analog reading = ");
    Serial.print(fsrR);

    if (fsrR < 10) {
        Serial.println(" - Tidak ada tekanan");
        stJ = "0";
    } else if (fsrR < 200) {
        Serial.println(" - Sentuhan ringan");
    }
}
```

```

stJ = "0";

} else if (fsrR < 500) {
    Serial.println(" - Tekanan ringan");
    stJ = "0";
} else if (fsrR < 800) {
    Serial.println(" - Tekanan sedang");
    stJ = "1";
} else {
    Serial.println(" - Tekanan berat");
    stJ = "1";
}

Serial.print("Status Jok = ");
Serial.println(stJ);
}

void MasukDataAcc(){
    while(!mpu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G)){
        Serial.println("Sensor MPU6050 tidak terdeteksi, cek pengkabelan.");
        delay(500);
    }
    mpu.setAccelPowerOnDelay(MPU6050_DELAY_3MS);//power supply delay selama 3ms
    mpu.setIntFreeFallEnabled(false);//mematikan hardware interrupt
    mpu.setIntZeroMotionEnabled(false);
    mpu.setIntMotionEnabled(false);
    mpu.setDHPFMode(MPU6050_DHPF_5HZ);//set high pass filter
    mpu.setMotionDetectionThreshold(1);//set limit deteksi pergerakan ke 4mg(dibagi 2)
    mpu.setMotionDetectionDuration(5);//set durasi minimal 5ms
    mpu.setZeroMotionDetectionThreshold(4);//set limit deteksi kediaman ke 8mg(dibagi 2)
    mpu.setZeroMotionDetectionDuration(2);//set durasi minimal 2s

    Vector rawAccel = mpu.readRawAccel();
    Activites act = mpu.readActivites();

```



```

if (act.isActivity)
{
    Serial.println ("GERAKAN MOTOR TERDETEKSI");
    stP = "1";
    Serial.print("Status Pergerakan = ");
    Serial.println(stP);
} else
{
    Serial.println ("NORMAL");
    stP = "0";
    Serial.print("Status Pergerakan = ");
    Serial.println(stP);
}
delay(1000);
}

void AlarmONOFF(){
    while(stK&&stJ=="0"){
        if(stP=="1"){
            StopAlarm();
        }
        else{
            StartAlarm();
        }
    }

    while(stK=="0"){
        if(stJ=="1"){
            StartAlarm();
        }
        else{
            StopAlarm();
        }
    }
}

```

```
while(stK=="1"){
    StartAlarm();
}

SIM800LREQUEST_Alarm();

if(stALst=="1"){
    StartAlarm();
}
else if(stALst=="0"){
    StopAlarm();
    stK="0";
}
}

void StartAlarm() {
    tone(11,450);

    gprsSerial.println("AT");
    ShowResponse();

    gprsSerial.println("AT+CPIN?"); //sets the password of the mobile device
    ShowResponse();

    gprsSerial.println("AT+CREG?"); //gives info abt registration status
    ShowResponse();

    gprsSerial.println("AT+CGATT?"); //attach or detach device to packet domain service
    ShowResponse();

    gprsSerial.println("AT+CIPSHUT"); //close GPRS PDP context
    ShowResponse();
}
```



```
gprsSerial.println("AT+CIPSTATUS"); //returns the corrent connection status

ShowResponse();

gprsSerial.println("AT+CIPMUX=0"); //single IP connection

ShowResponse();

gprsSerial.println("AT+CSTT=\"internet\""); //sets up the apn, user name and password for
the PDP context.

ShowResponse();

gprsSerial.println("AT+CIICR"); //brings up the GPRS or CSD call depending on the
configuration previously set by the AT+CSTT command.

ShowResponse();

gprsSerial.println("AT+CIFSR"); //returns the local IP addres

ShowResponse();

gprsSerial.println("AT+CIPSPRT=0"); //set whether echo prompt ">" after issuing
"AT+CIPSEND" command.

ShowResponse();

gprsSerial.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \"80\"); //starts TCP or
UDP connection

ShowResponse();

gprsSerial.println("AT+CIPSEND"); //send the data over the TCP or UDP connection.

ShowResponse();

String alarmON="GET
https://api.thingspeak.com/update?api_key=VNBV9VJOHRXJL2UL&field6=,1";

Serial.println(alarmON);

gprsSerial.println(alarmON); //begin send data to remote server

ShowResponse();

gprsSerial.println((char)26); //sending
```

```

delay(5000); //waiting for reply, important! the time is base on the condition of internet

gprsSerial.println();
ShowResponse();

gprsSerial.println("AT+CIPSHUT"); //close the GPRS PDP context.
ShowResponse();

//KIRIM SMS KE PONSEL PENGGUNA
gprsSerial.println("AT+CMGF=1"); //set format pesan SMS sebagai text
ShowResponse();
gprsSerial.println("AT+CMGS=\"+6281318690931\""); //nomor ponsel pengguna sistem
ShowResponse();
gprsSerial.println("PERINGATAN! Ada aktivitas mencurigakan di motor Anda!"); //isi SMS
ShowResponse();
gprsSerial.write(26);
}

void StopAlarm() {
    noTone(11);

    gprsSerial.println("AT");
    ShowResponse();

    gprsSerial.println("AT+CPIN?"); //sets the password of the mobile device
    ShowResponse();

    gprsSerial.println("AT+CREG?"); //gives info abt registration status
    ShowResponse();

    gprsSerial.println("AT+CGATT?"); //attach or detach device to packet domain service
    ShowResponse();

    gprsSerial.println("AT+CIPSHUT"); //close GPRS PDP context

```



```
ShowResponse();

gprsSerial.println("AT+CIPSTATUS"); //returns the corrent connection status
ShowResponse();

gprsSerial.println("AT+CIPMUX=0"); //single IP connection
ShowResponse();

gprsSerial.println("AT+CSTT=\"internet\""); //sets up the apn, user name and password for
the PDP context.
ShowResponse();

gprsSerial.println("AT+CIICR"); //brings up the GPRS or CSD call depending on the
configuration previously set by the AT+CSTT command.
ShowResponse();

gprsSerial.println("AT+CIFSR"); //returns the local IP addres
ShowResponse();

gprsSerial.println("AT+CIPSPRT=0"); //set whether echo prompt ">" after issuing
"AT+CIPSEND" command.
ShowResponse();

gprsSerial.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \"80\"); //starts TCP or
UDP connection
ShowResponse();

gprsSerial.println("AT+CIPSEND"); //send the data over the TCP or UDP connection.
ShowResponse();

String alarmOFF="GET
https://api.thingspeak.com/update?api_key=VNBV9VJOHRXJL2UL&field6=,0";

Serial.println(alarmOFF);

gprsSerial.println(alarmOFF); //begin send data to remote server
ShowResponse();
```

```

gprsSerial.println((char)26);//sending
    delay(5000);//waitting for reply, important! the time is base on the condition of internet
    gprsSerial.println();
    ShowResponse();

    gprsSerial.println("AT+CIPSHUT");//close the GPRS PDP context.
    ShowResponse();
}

void SIM800LSEND(){
    gprsSerial.println("AT");
    ShowResponse();

    gprsSerial.println("AT+CPIN?");//sets the password of the mobile device
    ShowResponse();

    gprsSerial.println("AT+CREG?");//gives info abt registration status
    ShowResponse();

    gprsSerial.println("AT+CGATT?");//attach or detach device to packet domain service
    ShowResponse();

    gprsSerial.println("AT+CIPSHUT");//close GPRS PDP context
    ShowResponse();

    gprsSerial.println("AT+CIPSTATUS");//returns the corrent connection status
    ShowResponse();

    gprsSerial.println("AT+CIPMUX=0");//single IP connection
    ShowResponse();

    gprsSerial.println("AT+CSTT=\"internet\"");//sets up the apn, user name and password for
    the PDP context.

```



```
ShowResponse();

gprsSerial.println("AT+CIICR");//brings up the GPRS or CSD call depending on the
configuration previously set by the AT+CSTT command.

ShowResponse();

gprsSerial.println("AT+CIFSR");//returns the local IP address

ShowResponse();

gprsSerial.println("AT+CIPSPRT=0");//set whether echo prompt ">" after issuing
"AT+CIPSEND" command.

ShowResponse();

gprsSerial.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \"80\");//starts TCP or
UDP connection

ShowResponse();

gprsSerial.println("AT+CIPSEND");//send the data over the TCP or UDP connection.

ShowResponse();

String str="GET https://api.thingspeak.com/update?api_key=VNBV9VJOHRXJL2UL&field1="
+ stK + "&field2=" + stJ + "&field3=" + stP + "&field4=" + Lat + "," + Lng;

Serial.println(str);

gprsSerial.println(str);//begin send data to remote server

ShowResponse();

gprsSerial.println((char)26);//sending
delay(5000);//waiting for reply, important! the time is base on the condition of internet
gprsSerial.println();

ShowResponse();

gprsSerial.println("AT+CIPSHUT");//close the GPRS PDP context.

ShowResponse();
}
```

```

Serial.print("Status Alarm :");
Serial.println(stALst);

gprsSerial.println("AT+HTTPTERM");
ShowResponse();

gprsSerial.println("AT+SAPBR=0,1");
ShowResponse();
}

GetResponse(5000); // respon di simpan kedalam variabel dataMasuk

Serial.print("Full data :");
Serial.println(datAMsk);

stALst = parse(datAMsk, ',', 1);

gprsSerial.print("AT+HTTPPARA=\"URL\\\",\\\"http://api.thingspeak.com/channels/1070612/
fields/6/last.txt\"");
gprsSerial.println("");
ShowResponse();

// set http action type 0 = GET, 1 = POST, 2 = HEAD
gprsSerial.println("AT+HTTPACTION=0");
GetResponse(15000);

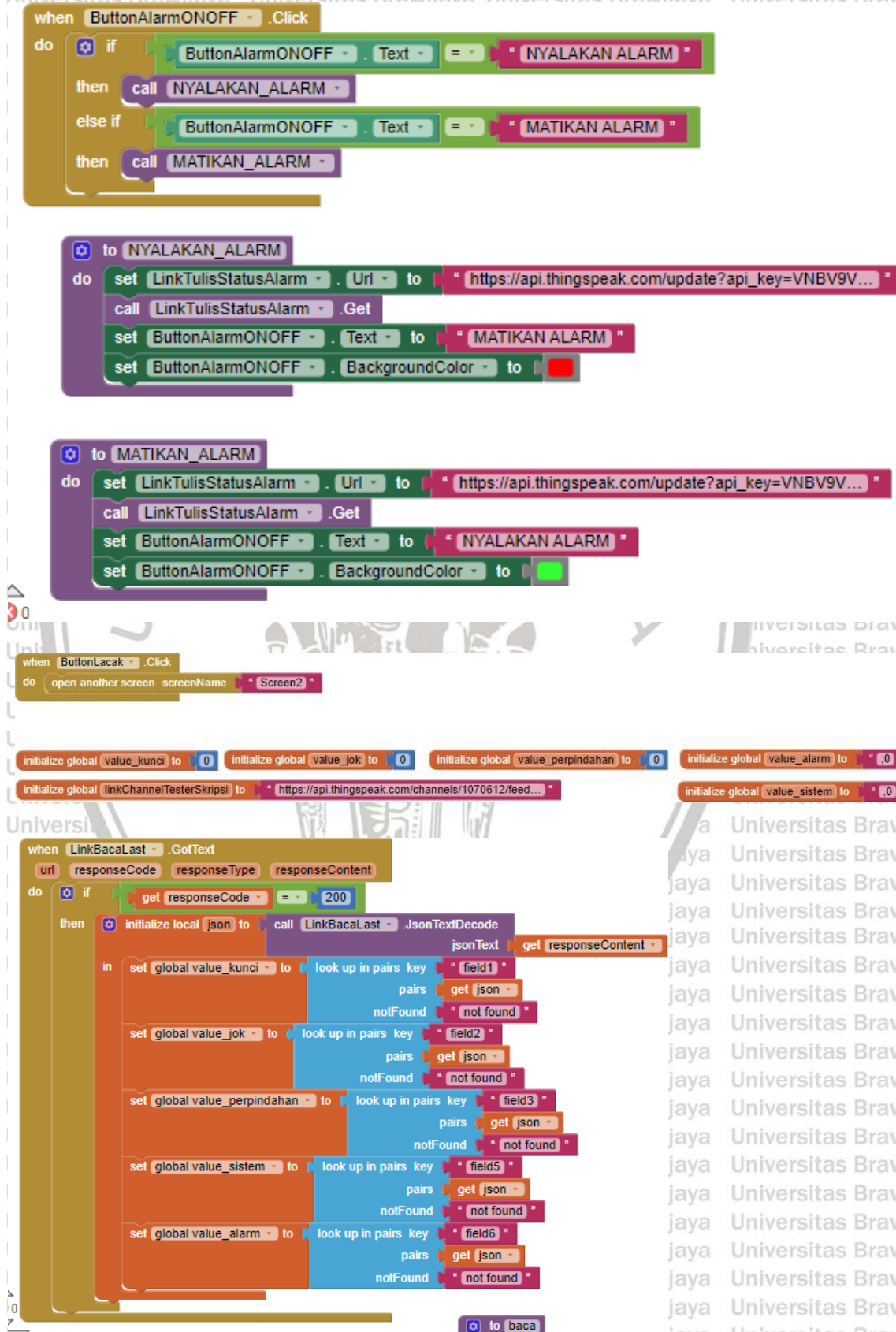
gprsSerial.println("AT+HTTPREAD=0");
GetResponse(5000); // respon di simpan kedalam variabel dataMasuk

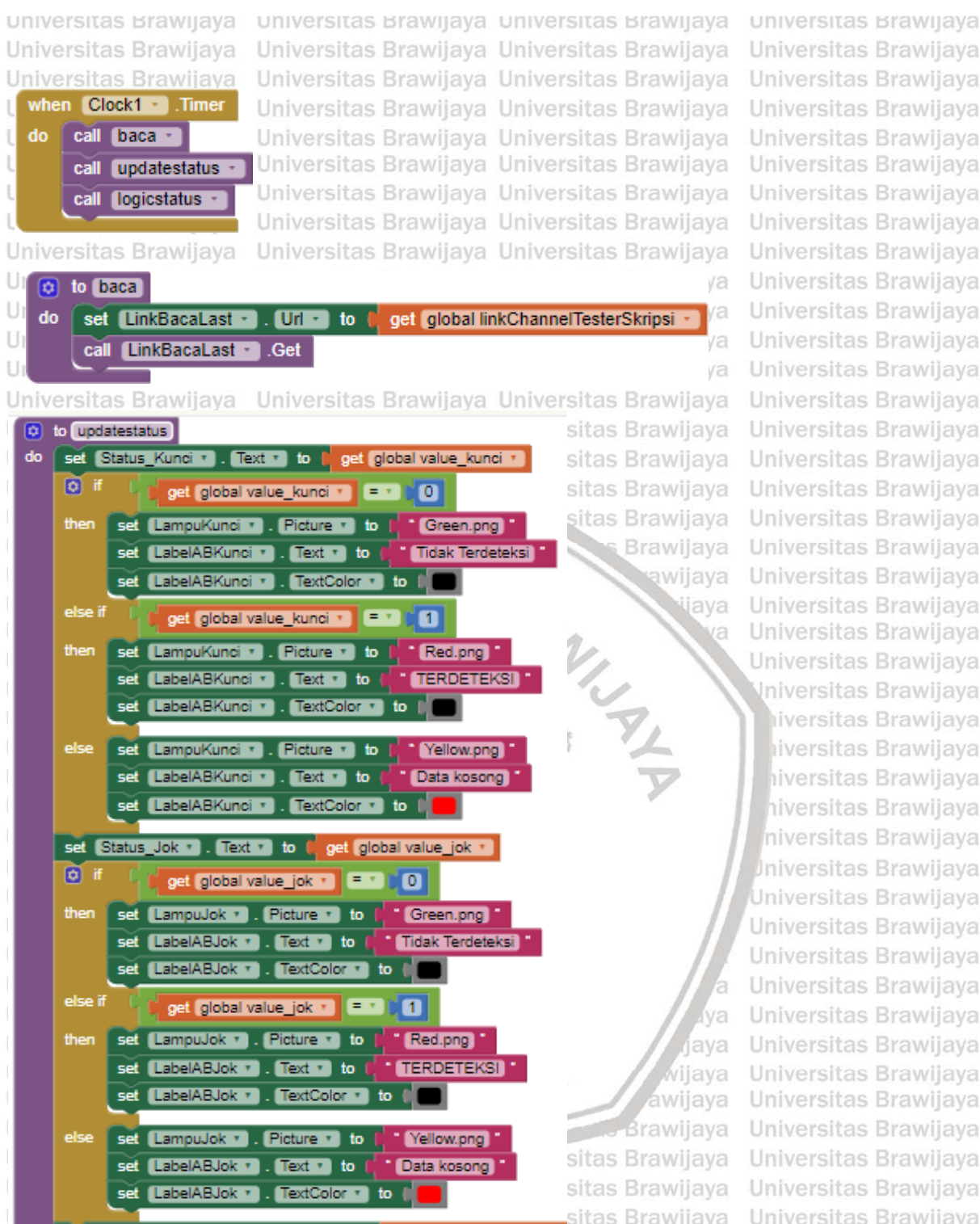
Serial.print("Full data :");
Serial.println(datAMsk);

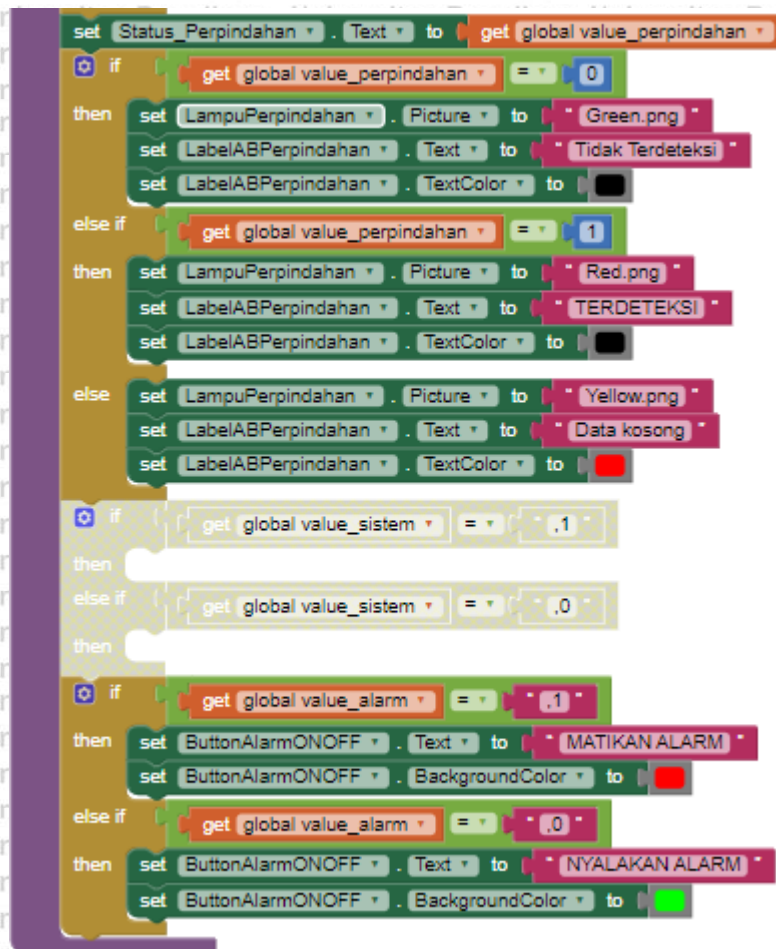
stALst = parse(datAMsk, ',', 1);

```


Lampiran 4. Blok pemrograman pada MIT App Inventor: Screen 1







Lampiran 5. Blok Pemrograman pada MIT App Inventor: Screen 2

